Process Expert - General Purpose Library

Process Templates Reference Manual

03/2023

EIO0000004043.06

www.se.com

Legal Information

The Schneider Electric brand and any trademarks of Schneider Electric SE and its subsidiaries referred to in this guide are the property of Schneider Electric SE or its subsidiaries. All other brands may be trademarks of their respective owners.

This guide and its content are protected under applicable copyright laws and furnished for informational use only. No part of this guide may be reproduced or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), for any purpose, without the prior written permission of Schneider Electric.

Schneider Electric does not grant any right or license for commercial use of the guide or its content, except for a non-exclusive and personal license to consult it on an "as is" basis. Schneider Electric products and equipment should be installed, operated, serviced, and maintained only by qualified personnel.

As standards, specifications, and designs change from time to time, information contained in this guide may be subject to change without notice.

To the extent permitted by applicable law, no responsibility or liability is assumed by Schneider Electric and its subsidiaries for any errors or omissions in the informational content of this material or consequences arising out of or resulting from the use of the information contained herein.

As part of a group of responsible, inclusive companies, we are updating our communications that contain non-inclusive terminology. Until we complete this process, however, our content may still contain standardized industry terms that may be deemed inappropriate by our customers.

© 2023 – Schneider Electric. All rights reserved.

Table of Contents

Safety Information 23
Qualification of Personnel 23
Proper Use 24
Before You Begin 24
Start-up and Test 25
Operation and Adjustments 25
About the Book 27
Overview 31
List of Process Templates 32
List of Process Families 32
Glossary of Acronyms 34
Acronym and Abbreviation 34
Template 36
Process Templates 36
Composition Strategy for Application Templates 37
Naming Convention 39
Common Parameters 43
Common Control Module Template Interfaces 47
Hyperlink Services 50
Control 52
Modularity 52
Function Block Interface 52
Supervision 55
Supervision Services 55
Supervision Resource Structure 55
Access Control 56
Privilege Levels 56
Calculated Variable 58
Calculated Variable 58
Commonly Used Items 59
Commonly Used Items 59
Genies 61
Using Genies 62
Genie Icons 63
Displaying Genies 65
Faceplate 65
Tab Icons 66
Alarm Tab 67
Event Tab 69
Trend Tab 69
Measures Tab 70
Color Codes 71
Dynamic Variable Status 71
Optional Services 73
Failures 74
Template 74
Description 74
Parameters 74

Control 75
DFB Representation 75
Inputs 76
Outputs 76
Inputs/Outputs 77
Public Variables 78
Supervision 78
Failures Tab 78
Interlocks 81
Template 81
..... 81
Control 82
Description 82
DFB Representation 83
Inputs 83
Outputs 84
Inputs/Outputs 85
Public Variables 86
Supervision 87
Interlocks Tab 87
Maintenance 89
Template 89
..... 89
Control 90
Description 90
DFB Representation 90
Inputs 90
Outputs 91
Inputs/Outputs 91
Public Variables 92
Supervision 92
Maintenance 92
Template - Alternate Services (Advance Motor Maintenance) 93
Maintenance 93
Control - Alternate Services (Advance Motor Maintenance) 94
Description 94
DFB Representation 94
Inputs 95
Outputs 95
Inputs/Outputs 95
Public Variables 96
Supervision - Alternate Services (Advance Motor Maintenance) 97
Items 97
Faceplate 99
Template - Alternate Services (Advance Valve Maintenance) 101
Maintenance 101
Control - Alternate Services (Advance Valve Maintenance) 101
Description 101
DFB Representation 102
Inputs 102
Outputs 102

Inputs/Outputs 102
Public Variables 103
Supervision - Alternate Services (Advance Valve Maintenance) 104
Items 104
Faceplate 106
Analog Alarm 108
Template 108
Description 108
Parameters 109
Control 110
DFB Representation 110
Inputs 111
Outputs 112
Inputs/Outputs 112
Public Variables 113
Supervision 114
Faceplate 114
Items 116
Template - Alternate Service (LALARM) 118
Description 118
Parameters 118
Control - Alternate Service (LALARM) 120
DFB Representation 120
Inputs 120
Outputs 121
Inputs/Outputs 121
Public Variables 122
Supervision - Alternate Services (LALARM) 123
Items 123
Template - Alternate Service (DALARM) 124
Description 124
Parameters 124
Control - Alternate Service (DALARM) 126
DFB Representation 126
Inputs 126
Outputs 127
Inputs/Outputs 127
Public Variables 128
Supervision - Alternate Services (DALARM) 128
Items 128
Template - Alternate Service (RALARM) 130
Description 130
Parameters 130
Control - Alternate Service (RALARM) 131
DFB Representation 131
Inputs 131
Outputs 132
Inputs/Outputs 132
Public Variables 133
Supervision - Alternate Services (RALARM) 133
Items 133

Group Control Link 135
Motors 137
Motor 138
Template 138
Description 138
Parameters 139
Interfaces 142
Control 143
DFB Representation 143
Inputs 143
Outputs 144
Inputs/Outputs 146
Public Variables 148
Optional Service - Local Panel (MOTORLPGP) 150
DFB Representation 150
Inputs 150
Outputs 151
Inputs/Outputs 152
Public Variables 153
Supervision 154
Genies 154
Faceplate 155
Items 157
Dual Speed and Dual Direction Motor 162
Template 162
Description 162
Parameters 163
Interfaces 167
Template - $Motor2SpdGP 167
Description 167
Parameters 168
Interfaces 172
Control 172
DFB Representation 173
Inputs 173
Outputs 175
Inputs/Outputs 178
Public Variables 179
Optional Service - Local Panel (MOTOR2LPGP) 182
DFB Representation 182
Inputs 182
Outputs 184
Inputs/Outputs 184
Public Variables 185
Supervision 186
Genies 186
Faceplate 187
Items 191
Variable Speed Motor 197
Template 197
Description 197

Parameters 198
Interfaces 202
Control 203
DFB Representation 203
Inputs 203
Outputs 205
Inputs/Outputs 208
Public Variables 210
Optional Service - Local Panel (MOTORVSLPGP) 212
Description 212
DFB Representation 213
Inputs 214
Outputs 215
Inputs/Outputs 216
Public Variables 217
Supervision 218
Genies 218
Faceplate 219
Items 222
Valves 228
Valve 229
Template 229
Description 229
Parameters 230
Interfaces 233
Control 233
DFB Representation 234
Inputs 234
Outputs 235
Inputs/Outputs 237
Public Variables 239
Optional Service - Local Panel (VALVELPGP) 240
DFB Representation 241
Inputs 241
Outputs 242
Inputs/Outputs 243
Public Variables 244
Supervision 244
Genies 245
Faceplate 247
Items 249
Motorized Control Valve 254
Template 254
Description 254
Parameters 255
Interfaces 259
Control 260
DFB Representation 260
Inputs 260
Outputs 262
Inputs/Outputs 266

Public Variables 268
Optional Service - Local Panel (MVALVELPGP) 270
DFB Representation 270
Inputs 270
Outputs 272
Inputs/Outputs 273
Public Variables 274
Supervision 274
Genies 274
Faceplates 275
Items 278
Hand Valve 283
Template 283
Description 283
Interfaces 284
Parameters 284
Control 285
DFB Representation 285
Inputs 286
Outputs 286
Inputs/Outputs 286
Public Variables 287
Supervision 288
Genies 288
Faceplate 289
Items 291
Dual Output Valve 294
Template 294
Description 294
Parameters 295
Interfaces 299
Control 299
DFB Representation 300
Inputs 300
Outputs 302
Inputs/Outputs 305
Public Variables 306
Optional Service - Local Panel (DVALVELPGP) 309
DFB Representation 309
Inputs 309
Outputs 311
Inputs/Outputs 311
Public Variables 312
Supervision 313
Genies 313
Faceplate 315
Items 318
Motorized Valve 324
Template 324
Description 324
Parameters 325

Interfaces 328
Control 329
DFB Representation 329
Inputs 329
Outputs 331
Inputs/Outputs 334
Public Variables 335
Optional Service - Local Panel (MVALVEDLPGP) 338
DFB Representation 338
Inputs 338
Outputs 340
Inputs/Outputs 341
Public Variables 342
Supervision 342
Genies 342
Faceplate 343
Items 346
Control Valve 352
Template 352
Description 352
Parameters 353
Interfaces 356
Control 356
DFB Representation 357
Inputs 357
Outputs 358
Inputs/Outputs 359
Public Variables 360
Optional Service - Local Panel (CVALVELPGP) 362
Description 362
DFB Representation 363
Inputs 363
Outputs 365
Inputs/Outputs 366
Public Variables 367
Supervision 367
Genies 367
Faceplate 369
Items 372
Signals 377
Analog Input 378
Template 378
Description 378
Parameters 379
Interfaces 381
Control 381
DFB Representation 382
Inputs 382
Outputs 383
Inputs/Outputs 383
Public Variables 384

Supervision 385
Genies 385
Faceplate 388
Items 392
Analog Output 395
Template 395
Description 395
Parameters 396
Interfaces 398
Control 399
DFB Representation 399
Inputs 399
Outputs 400
Inputs/Outputs 401
Public Variables 402
Optional Service - Analog Output Local Panel (AOUTPUTLPGP) 403
DFB Representation 403
Inputs 404
Outputs 405
Inputs/Outputs 406
Public Variables 407
Supervision 407
Genies 407
Faceplate 409
Items 411
Digital Input 416
Template 416
Description 416
Parameters 417
Interfaces 419
Control 419
DFB Representation 419
Inputs 420
Outputs 420
Inputs/Outputs 421
Public Variables 421
Supervision 422
Genies 422
Faceplate 423
Items 425
Digital Output 428
Template 428
Description 428
Parameters 429
Interfaces 431
Control 431
DFB Representation 432
Inputs 432
Outputs 432
Inputs/Outputs 433

Public Variables 434
Supervision 435
Genies 435
Faceplate 436
Items 438
Range 441
Template 441
Description 441
Parameters 441
Interfaces 442
Totalizer 443
Template 443
Description 443
Parameters 444
Interfaces 446
Control 447
DFB Representation 447
Inputs 448
Outputs 450
Inputs/Outputs 452
Public Variables 453
Supervision 455
Genies 455
Faceplate 456
Items 458
Multiple Analog Input 462
Template 462
Description 462
Parameters 463
Interfaces 465
Control 466
DFB Representation 466
Inputs 467
Outputs 468
Inputs/Outputs 469
Public Variables 471
Supervision 472
Genies 472
Faceplate 475
Items 478
Load cell 482
Template 482
Description 482
Parameters 483
Interfaces 485
Control 486
DFB Representation 486
Inputs 486
Outputs 488
Inputs/Outputs 490
Public Variables 494

Diagnostic Information Management 495
Configuration of PMESWT Module 496
PMESWT DFB Work Flow 501
Supervision 503
Genies 503
Faceplate 504
Items 510
BMXERT1604 516
Supervision 516
Tags 516
Continuous Control 518
PID 519
Template 519
Description 519
Parameters 520
Interfaces 523
Control - PIDCTLGP 524
DFB Representation 524
Inputs 524
Outputs 525
Inputs/Outputs 526
Public Variables 528
Control - PIDMUXGP 529
DFB Representation 530
Inputs 530
Outputs 530
Inputs/Outputs 530
Public Variables 532
Control - PIDTUNEGP 532
DFB Representation 532
Outputs 533
Inputs/Outputs 533
Supervision 536
Genies 536
Faceplate 540
Items 544
Ratio Controller Management 550
Template 550
Description 550
Parameters 551
Interfaces 553
Control – RATIOCTLGP 554
DFB Representation 554
Inputs 554
Outputs 555
Inputs/Outputs 555
Public Variables 556
Control – ADVCTLDATAR 557
Description 557
DFB Representation 558
Inputs 558

Inputs/Outputs 558
Supervision 558
Genies 559
Faceplates 559
Items 561
Ramp 564
Template 564
Description 564
Parameters 565
Interfaces 566
Control 567
DFB Representation 567
Inputs 567
Outputs 568
Inputs/Outputs 569
Public Variables 570
Supervision 571
Genies 571
Faceplate 571
Items 573
Lead-Lag 576
Template 576
Description 576
Parameters 577
Interfaces 579
Control 579
DFB Representation 580
Inputs 580
Outputs 580
Inputs/Outputs 581
Public Variables 582
Supervision 583
Genies 583
Faceplate 584
Items 586
Split-Range Controller 589
Template 589
Description 589
Parameters 590
Interfaces 591
Control 592
DFB Representation 592
Inputs 593
Outputs 593
Inputs/Outputs 594
Public Variables 595
Supervision 596
Genies 596
Faceplate 596
Items 598
Step3 Controller 602

Template 602
Description 602
Parameters 603
Interfaces 604
Control 605
DFB Representation 606
Inputs 606
Outputs 607
Inputs/Outputs 608
Public Variables 609
Supervision 610
Genies 610
Faceplate 611
Items 614
Internal Model Controller 617
Template 617
Description 617
Parameters 618
Interfaces 620
Control 621
DFB Representation 621
Inputs 622
Outputs 623
Inputs/Outputs 623
Public Variables 625
Supervision 626
Genies 626
Faceplate 629
Items 634
Pulse Width Modulation Controller 638
Template 638
Description 638
Parameters 638
Interfaces 640
Control 641
DFB Representation 641
Inputs 642
Outputs 642
Inputs/Outputs 643
Public Variables 644
Supervision 645
Genies 645
Faceplate 645
Items 648
Sequential Control 651
Sequential Control 652
Description of Sequential Control 652
Definition of Status and Commands 653
Status Matrix 655
Status Controller 656
Operating Modes 656

Using the $SequenceDFB Template 657
SEQCTLGP - Sequential Control 662
Template 662
Description 662
Parameters 662
Interfaces 668
Control - SEQCTLGP 669
Description 669
DFB Representation 669
Inputs/Outputs 669
Public Variables 671
Supervision 674
Genies 674
Faceplate 675
Items 676
SEQCTLGP - Example of Use 680
Description 680
DFB Representation 681
Reusable Sequence Logic 681
Auxiliary Functions of Sequential Control 686
Control – SEQTRANSGP 686
Description 686
Inputs 687
Inputs/Outputs 688
Control – SEQFAILEDGP 688
Description 688
Inputs/Outputs 689
Control – SEQSTABLESTEPGP 689
Description 689
Inputs/Outputs 690
Control – SEQSTEPDESCGP 690
Description 690
Inputs 691
Inputs/Outputs 692
Public Variables 692
SEQPARxxGP - Sequence Parameter Management 693
Control 693
Description 693
DFB Representation 694
Inputs 694
Outputs 694
Inputs/Outputs 695
Public Variables 696
USERSEQxxGP - User Sequence Parameter 697
Control 697
Description 697
DFB Representation 698
Inputs 698
Inputs/Outputs 698
Auxiliary Functions 699
ASELECTGP - Analog Signal Selector 700

Template 700
Description 700
Parameters 701
Interfaces 703
Control 704
DFB Representation 704
Inputs 704
Outputs 705
Inputs/Outputs 705
Public Variables 706
Supervision 707
Supervision Functions 707
Genies 708
Faceplate 708
Items 709
Message Box 712
Template 712
Description 712
Parameters 713
Interfaces 714
Control 714
DFB Representation 714
Inputs 715
Outputs 715
Inputs/Outputs 715
Public Variables 717
Supervision 719
Genies 719
Faceplate 719
Items 720
Alarm Summary 722
Template 722
Description 722
Parameters 722
Interfaces 724
Control 724
Description 724
Supervision 724
Supervision Functions 725
Genies 725
Faceplate 725
Items 727
Bool Setpoint 728
Template 728
Description 728
Parameters 728
Interfaces 729
Supervision 729
Genies 729
Items 730
Real Setpoint 731

Template 731
Description 731
Parameters 731
Interfaces 732
Supervision 732
Genies 732
Items 733
Integer Setpoint 734
Template 734
Description 734
Parameters 734
Interfaces 735
Supervision 735
Genies 735
Items 736
Duration Setpoint 737
Template 737
Description 737
Parameters 737
Interfaces 738
Supervision 738
Genies 738
Items 739
Scheduler Function 740
Template 740
Description 740
Parameters 740
Interfaces 742
Control 742
DFB Representation 742
Inputs 743
Outputs 744
Inputs/Outputs 744
Supervision 745
Genies 745
Faceplate 746
Items 748
Calculations 750
Digital Calculation 751
Template 751
Description 751
Parameters 751
Interfaces 752
Control 752
DFB Representation 752
Inputs 753
Outputs 753
Public Variables 754
Analog Calculation 755
Template 755
Description 755

Parameters 756
Interfaces 757
Control 757
DFB Representation 757
Inputs 757
Outputs 758
Public Variables 759
Analog Linear 761
Template 761
Description 761
Parameters 761
Interfaces 762
Control 762
DFB Representation 762
Inputs 763
Outputs 763
Equipment Module 764
Pump Set Equipment Module 765
Description of Pump Set Equipment Module 765
Strategy Selection and Functionality 766
Disabling the Strategies Based on Equipment Module State 767
Input and Output Parameters 768
Input Parameters Applicable For Each Strategy 770
Pump Set Equipment Module State Machine 770
Pump Set Equipment Module Diagnosis State 771
Pump Set Pumping Asset Diagnosis State 772
Modifying the PumpSetPattern Template 772
Pump Set 785
Template 785
Description 785
Parameters 786
Interfaces 791
Control — PUMPSETPATTERNGP 791
Description 791
DFB Representation 792
Inputs 792
Outputs 793
Inputs/Outputs 793
Control — PUMPINGASSET05GP 797
DFB Representation 798
Outputs 798
Inputs/Outputs 798
Supervision 799
Genies 799
Faceplate 800
Items 802
Auxiliary Functions of Pump Set Equipment Module 804
PUMPINGASSETSEQGP 804
Description 804
DFB Representation 805
Inputs 806

Outputs 806
Inputs/Outputs 806
Flow Control Equipment Module 807
Description of Flow Control Equipment Module 807
Strategy Selection and Functionality 808
Disabling the Strategies Based on Equipment Module State 809
Input and Output Parameters 809
Input Parameters Applicable For Each Strategy 812
Flow Control Equipment Module State Machine 812
Setpoint Management 812
Flow Control Equipment Module Diagnosis State 814
Flow Control Pumping Asset Diagnosis State 814
Modifying the `FlowCtlPattern` Template 815
Flow Control 817
Template 817
Description 817
Parameters 817
Interfaces 822
Control — `FLOWCTLPATTERNGP` 823
Description 823
DFB Representation 823
Inputs 823
Outputs 825
Inputs/Outputs 825
Control — `FLOWASSET05GP` 829
DFB Representation 830
Outputs 830
Inputs/Outputs 830
Supervision 831
Genies 831
Faceplate 832
Items 834
Auxiliary Functions of Flow Control Equipment Module 836
`PUMPINGASSETSEQ1GPGP` 836
Description 836
DFB Representation 837
Inputs 838
Outputs 838
Inputs/Outputs 838
Equipment Module 839
Description of Equipment Module 839
Definition of Statuses, Equipment Module State and Commands 841
Strategy Execution Status Matrix 843
Strategy Execution Status Controller 844
Operating Modes 844
Modifying the `EMPattern` Template 845
`EMCTLGP` - Equipment Module Controller 852
Template 852
Description 852
Parameters 852

Interfaces 855
Control 855
Description 855
DFB Representation 856
Inputs 856
Inputs/Outputs 856
Public Variables 859
Supervision 862
Genies 862
Faceplate 863
Items 867
EMCTLGP - Example of Use 871
Description 871
DFB Representation 872
Reusable Equipment Module Logic 872
EMLOGICGP - User Defined Process Logic 881
Control 881
Description 881
DFB Representation 881
Inputs 882
Inputs/Outputs 882
Auxiliary Functions of Equipment Module 883
EMTRANSGP 883
Description 883
Inputs 884
Inputs/Outputs 885
EMFAILEDGP 885
Description 885
Inputs/Outputs 886
EMSTABLESTEPGP 886
Description 886
Inputs/Outputs 887
EMSTEPDESCGP 887
Description 887
Inputs 888
Inputs/Outputs 889
Public Variables 889
Phase Manager 890
PHASEGP - Phase Manager 891
Template 891
$PhaseGP - Phase Manager 891
Parameters 892
Interfaces 895
Control 896
Description of Phase Manager 896
Definition of Phase Statuses, Phase Statuses, Phase Commands, and Phase Commands 897
Status Matrix 899
Status Controller 900
Operating Modes 901
DFB Representation 901

Inputs 901
Inputs/Outputs 902
Supervision 906
Genies 906
Faceplate 912
Items for InBatch Legacy 918
Items for InBatch and Recipe Management 923
Auxiliary Functions of PHASEGP 930
PHTRANSGP 930
Description 930
Inputs 931
Inputs/Outputs 932
PHFAILEDGP 932
Description 932
Inputs/Outputs 933
PHSTABLESTEPGP 933
Description 933
Inputs/Outputs 934
PHSTEPDESCGP 934
Description 934
Inputs 935
Inputs/Outputs 936
Public Variables 936
PHPARxxGP 937
Description 937
DFB Representation 938
Inputs 938
Outputs 939
Inputs/Outputs 939
Public Variables 940
UserPhasexxGP 942
DFB Representation 942
Inputs 942
Inputs/Outputs 943
PHASEGP - Example of Use 944
Description 944
DFB Representation 945
Reusable Phase Logic 945
Advanced Process Control 951
Advanced PID 952
Template 952
Description 952
Parameters 953
Interfaces 953
Control – PIDADVCTLGP 954
DFB Representation 954
Inputs 954
Outputs 955
Inputs/Outputs 955
Public Variables 957
Control – PIDMUXGP 958

Control – PIDTUNEGP 958
Control – ADVCTLDATA 958
Description 958
DFB Representation 959
Inputs 959
Inputs/Outputs 960
Supervision 961
Genies 961
Faceplate 961
Items 962
Two Loop PID Control 968
Template 968
Description 968
Parameters 968
Interfaces 969
Control 970
Supervision 970
Genies 971
Faceplate 972
Items 974
Four Loop PID Control 975
Template 975
Description 975
Parameters 976
Interfaces 977
Control 978
Supervision 978
Genies 978
Faceplate 980
Items 982
Index 983

Safety Information

Important Information

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

⚠ CAUTION

CAUTION indicates a hazardous situation which, if not avoided, **could result** in minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

Please Note

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

Qualification of Personnel

A qualified person is one who has the following qualifications:

- Skills and knowledge related to the construction and operation of electrical equipment and the installation.
- Knowledge and experience in industrial control programming.
- Received safety-related training to recognize and avoid the hazards involved.

The qualified person must be able to detect possible hazards that may arise from parameterization, modifying parameter values and generally from mechanical,

electrical, or electronic equipment. The qualified person must be familiar with the standards, provisions, and regulations for the prevention of industrial accidents, which they must observe when designing and implementing the system.

Proper Use

This product is a library to be used together with the automation control systems and is intended solely for the purposes described in the present documentation as applied in the industrial sector.

Always observe the applicable safety-related instructions, the specified conditions, and the technical data.

Perform a risk evaluation concerning the specific use before using the product. Take protective measures according to the result.

Since the product is used as a part of an overall system, you must ensure the safety of the personnel by means of the concept of this overall system (for example, machine concept).

Any other use is not intended and may be hazardous.

Before You Begin

Do not use this product on machinery lacking effective point-of-operation guarding. Lack of effective point-of-operation guarding on a machine can result in serious injury to the operator of that machine.

⚠ WARNING

UNGUARDED EQUIPMENT

- Do not use this software and related automation equipment on equipment which does not have point-of-operation protection.
- Do not reach into machinery during operation.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

This automation equipment and related software is used to control a variety of industrial processes. The type or model of automation equipment suitable for each application will vary depending on factors such as the control function required, degree of protection required, production methods, unusual conditions, government regulations, etc. In some applications, more than one processor may be required, as when backup redundancy is needed.

Only you, the user, machine builder or system integrator can be aware of all the conditions and factors present during setup, operation, and maintenance of the machine and, therefore, can determine the automation equipment and the related safeties and interlocks which can be properly used. When selecting automation and control equipment and related software for a particular application, you should refer to the applicable local and national standards and regulations. The National Safety Council's Accident Prevention Manual (nationally recognized in the United States of America) also provides much useful information.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the operator's hands and other parts of the body are free to enter the pinch points or other hazardous areas and serious injury can occur. Software products alone cannot protect an operator from injury. For this reason the software cannot be substituted for or take the place of point-of-operation protection.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before

placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of the Function Block Library, System User Guide, or other implementation referenced in this documentation.

Start-up and Test

Before using electrical control and automation equipment for regular operation after installation, the system should be given a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such a check are made and that enough time is allowed to perform complete and satisfactory testing.

⚠ WARNING

EQUIPMENT OPERATION HAZARD

- Verify that all installation and set up procedures have been completed.
- Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices.
- Remove tools, meters, and debris from equipment.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Follow all start-up tests recommended in the equipment documentation. Store all equipment documentation for future references.

Software testing must be done in both simulated and real environments.

Verify that the completed system is free from all short circuits and temporary grounds that are not installed according to local regulations (according to the National Electrical Code in the U.S.A, for instance). If high-potential voltage testing is necessary, follow recommendations in equipment documentation to prevent accidental equipment damage.

Before energizing equipment:

- Remove tools, meters, and debris from equipment.
- Close the equipment enclosure door.
- Remove all temporary grounds from incoming power lines.
- Perform all start-up tests recommended by the manufacturer.

Operation and Adjustments

The following precautions are from the NEMA Standards Publication ICS 7.1-1995:

(In case of divergence or contradiction between any translation and the English original, the original text in the English language will prevail.)

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly operated.

- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or unsafe operation. Always use the manufacturer's instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer's instructions and the machinery used with the electrical equipment.
- Only those operational adjustments required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.

About the Book

Document Scope

This document describes the functionality and features of the control services, Supervision services, and process templates.

This document only covers the functional aspects of templates, control, and Supervision services when engineering a system, using EcoStruxure™ Process Expert, and describes the dynamic objects visible from the runtime. It does not cover any operational aspects, nor does it provide information on how to use the Supervision services to monitor and operate control systems.

To use process templates, you need to have knowledge of EcoStruxure™ Process Expert, Supervision, and Control Participants.

Validity Note

This document has been updated for the release of EcoStruxure™ Process Expert 2021.

Related Documents

The characteristics that are described in the present document, as well as those described in the documents included in the Related Documents section below, can be found online. To access the information online, go to the Schneider Electric home page www.se.com/ww/en/download/.

The characteristics that are described in the present document should be the same as those characteristics that appear online. In line with our policy of constant improvement, we may revise content over time to improve clarity and accuracy. If you see a difference between the document and online information, use the online information as your reference.

Title of Documentation	Reference Number
EcoStruxure™ Process Expert - General Purpose Library User Guide	EIO0000004045

Technical Support

Visit https://www.se.com/myschneider/ for support, software updates, and latest information.

Product Related Information

⚠ WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate them.
- Review the implications of communication link interruptions and take actions to mitigate them.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, and applicable codes and regulations.
- Apply local accident prevention and safety regulations and guidelines.[1]
- Test each implementation of a system for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

Examples described in this manual are provided for information only.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Adapt examples that are given in this manual to the specific functions and requirements of your industrial application before you implement them.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Terminology Derived from Standards

The technical terms, terminology, symbols and the corresponding descriptions in this manual, or that appear in or on the products themselves, are generally derived from the terms or definitions of international standards.

In the area of functional safety systems, drives and general automation, this may include, but is not limited to, terms such as safety, safety function, safe state, fault, fault reset, malfunction, failure, error, error message, dangerous, etc.

Among others, these standards include:

Standard	Description
IEC 61131-2:2007	Programmable controllers, part 2: Equipment requirements and tests.
ISO 13849-1:2015	Safety of machinery: Safety related parts of control systems. General principles for design.
EN 61496-1:2013	Safety of machinery: Electro-sensitive protective equipment.

Standard	Description
	Part 1: General requirements and tests.
ISO 12100:2010	Safety of machinery - General principles for design - Risk assessment and risk reduction.
EN 60204-1:2006	Safety of machinery - Electrical equipment of machines - Part 1: General requirements.
ISO 14119:2013	Safety of machinery - Interlocking devices associated with guards - Principles for design and selection.
ISO 13850:2015	Safety of machinery - Emergency stop - Principles for design.
IEC 62061:2015	Safety of machinery - Functional safety of safety-related electrical, electronic, and electronic programmable control systems.
IEC 61508-1:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: General requirements.
IEC 61508-2:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Requirements for electrical/electronic/ programmable electronic safety-related systems.
IEC 61508-3:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Software requirements.
IEC 61784-3:2016	Industrial communication networks - Profiles - Part 3: Functional safety fieldbuses - General rules and profile definitions.
2006/42/EC	Machinery Directive
2014/30/EU	Electromagnetic Compatibility Directive
2014/35/EU	Low Voltage Directive

In addition, terms used in the present document may tangentially be used as they are derived from other standards such as:

Standard	Description
IEC 60034 series	Rotating electrical machines
IEC 61800 series	Adjustable speed electrical power drive systems
IEC 61158 series	Digital data communications for measurement and control – Fieldbus for use in industrial control systems

Finally, the term zone of operation may be used in conjunction with the description of specific hazards, and is defined as it is for a hazard zone or danger zone in the Machinery Directive (2006/42/EC) and ISO 12100:2010.

NOTE: The aforementioned standards may or may not apply to the specific products cited in the present documentation. For more information concerning the individual standards applicable to the products described herein, see the characteristics tables for those product references.

Overview

What's in This Part

List of Process Templates 32
Glossary of Acronyms.....34
Template 36
Control 52
Supervision.....55

Overview

This part explains in general the EcoStruxure Process Expert process library and concepts implemented for controlling process equipment.

List of Process Templates

What's in This Chapter

List of Process Families 32

List of Process Families

The process templates described in this document are grouped by family.

The table lists the control module templates of each family:

Family name	Process templates	Description
Motors	$MotorGP, page 138	Motor
	$Motor2DirGP, page 162	Motor two direction
	$Motor2SpdGP, page 162	Motor two speed
	$MotorVSGP, page 197	Motor variable speed
Valves	$ValveGP, page 229	Valve
	$MValvedWithPosGP, page 254	Motorized control valve with position
	$HandValveGP, page 283	Hand valve
	$DualOPValveGP, page 294	Dual actuator valve
	$MValveGP, page 324	Motorized valve
	$ControlValveGP, page 352	Control valve
Signals	$AnalogInputGP, page 378	Analog input
	$AnalogOutputGP, page 395	Analog output
	$DigitalInputGP, page 416	Digital input
	$DigitalOutputGP, page 428	Digital output
	$TotalGP, page 443	Total - totalizer
	$AnalogInMultiGP, page 462	Multiple analog input
	$LoadCellGP, page 482	Load cell
	$BMXERT1604, page 516	16 Channel digital input management
Continuous control	$PIDGP, page 519	PID control
	$RatioCtrlGP, page 550	Ratio control
	$RampGP, page 564	Ramp
	$LeadLagGP, page 576	Lead-Lag
	$SplitRangeGP, page 589	Split-range controller
	$Step3CtlGP, page 602	Step3 controller
	$IMCGP, page 617	Internal model controller
	$PWMOutputGP, page 638	Pulse width modulation controller
Sequential control	$SequenceGP, page 662	Sequential control
Phase Manager	$PhaseGP, page 891	Phase manager
Auxiliary functions	$AnalogSelectGP, page 700	Analog select process control

Family name	Process templates	Description
	$MessageBoxGP, page 712	Message box
	$AlaramSummaryGP, page 722	Alarm summary
	$SPBoolGP, page 728	Setpoint bool
	$SPRealGP, page 731	Setpoint real
	$SPIntGP, page 734	Setpoint integer
	$SPDurationGP, page 737	Setpoint duration
	$SchedulerGP, page 740	Scheduler function
Calculations	$DigitalCalcGP, page 751	Digital calculation
	$AnalogCalcGP, page 755	Analog calculation
	$AnalogLinearGP, page 761	Analog linear
Equipment module	$PumpSetPatternGP, page 785	Pumpset pattern
	$FlowCtlPatternGP, page 817	Flow pattern
	$EMPatternGP, page 852	Equipment pattern
Advanced process control	PIDADVGP, page 952	Advanced PID
	PID2LoopGP, page 968	Two loop PID control
	PID4LoopGP, page 975	Four loop PID control

Glossary of Acronyms

What's in This Chapter

Acronym and Abbreviation 34

Acronym and Abbreviation

The following table lists the acronyms, Abbreviation, and its definition used in this manual:

Acronym/Abbreviation	Definition
DFB	Derived function block
FBD	Function block diagram
EFB	Elementary function block
HMI	Human machine interface
PID	Proportional integral derivative
ZSS	Zero speed switch
MMM	Mining metals minerals
AO	Analog output
DO	Digital output
FwdCmd	Forward command
RevCmd	Reverse command
FwdRun	Forward run
RevRun	Reverse run
OpenCmd	Open command
CloseCmd	Close command
RunCmd	Run command
RunFB	Run feedback command
StopFB	Stop feedback command
OP	Output command
LOP	Local output command
PV	Present value
SP	Setpoint
LSP	Local setpoint
RSP	Remote setpoint
LSPSel	Select local setpoint
OOS	Out of service
Maint	Maintenance
ILCK	Interlock
ILCKBP	Interlock bypassed
RunningFB	Running feedback command
StoppedFB	Stopped feedback command
FwdILCK	Forward interlock
RevILCK	Reverse interlock
Sim	Simulation

Acronym/Abbreviation	Definition
Run1FB	Forward direction/speed 1
Run2FB	Reverse direction/speed 2
ILCKPos	Interlock position
ILCK1Pos	Interlock position of forward direction/speed 1
ILCK2Pos	Interlock position of reverse direction/speed 2
UnPos	Indeterminable position
ExtCTLD	Device controlled from external system
LOpenOP	Local open output command
LCloseOP	Local close output command
SCADA	Supervisory control and data acquisition
EPE	EcoStruxure process expert
PO	Peer owner
PC	Peer consumer

Template

What's in This Chapter

Process Templates 36
Composition Strategy for Application Templates 37
Naming Convention 39
Common Parameters 43
Common Control Module Template Interfaces 47
Hyperlink Services 50

Overview

This chapter describes the basic concepts of the process template.

Process Templates

Description

The EcoStruxure Process Expert General Purpose Library provides resources that have been pre-configured and tested by Schneider Electric, and that are designed for automating a large variety of processes. The EcoStruxure Process Expert General Purpose Library provides resources that have been pre-configured and tested by Schneider Electric, and that are designed for automating a large variety of processes.

Process templates model generic functionalities of process equipment but also hardwired signals from the field, and variables. They represent the application of the system.

They are used to implement the Control and Supervision services that are provided by the software participants, streamlining the engineering of systems.

The resources providing these services are encapsulated in dedicated facet references, which are then organized in composite references.

At the highest level, the process template represents the control module.

Using interfaces, you can also link process templates with other compatible device and communication templates.

Process Template Services

Facet templates referenced by process control module templates provide the following services:

Control	Includes core services plus additional, optional services, which you can activate if needed. Function blocks and variables are the resources that are encapsulated in these facet templates to provide such services.
Supervision	Supervision services complement the Control services. Supervision services are optional and those corresponding to selected Control are activated by default. Tags, alarms, and genies are the resources that are encapsulated in these facet references to provide such services. Data is provided by the associated Control resources.

Also, for both Control and Supervision services, you can configure parameters during instantiation to meet the requirements of your application.

Composition Strategy for Application Templates

General

The following general rules are applied when designing application library templates:

- Decoupling between participants
- Coupling into the participant
- Composition approach
- Exposed interfaces

Decoupling Between Participants

Maximum decoupling between facets from different participants is expected. One participant does not need to know the internal implementation of the other one to be able to collaborate.

For instance, the Control logic facets expose the OPC (PLC - HMI variable) items to be required for the Citect SCADA data facets in order that the Citect SCADA data facets are not altered as a result of any change in the Control logic facet data structures (DDT field names). So, the interfaces between participants contain so many items as required to respect this rule.

Coupling into the Participant

To minimize the data to be exchanged between facets from the same participant, the library designer can assume the details of the data structure that is shared between the facets by means of the interfaces.

For instance, the Citect SCADA Genie facets can assume which is the naming convention being implemented for Citect SCADA tags into the Citect SCADA data facets. That means that, by sharing the name of the object being implemented through the interface, the genie can assume the name of tags involved.

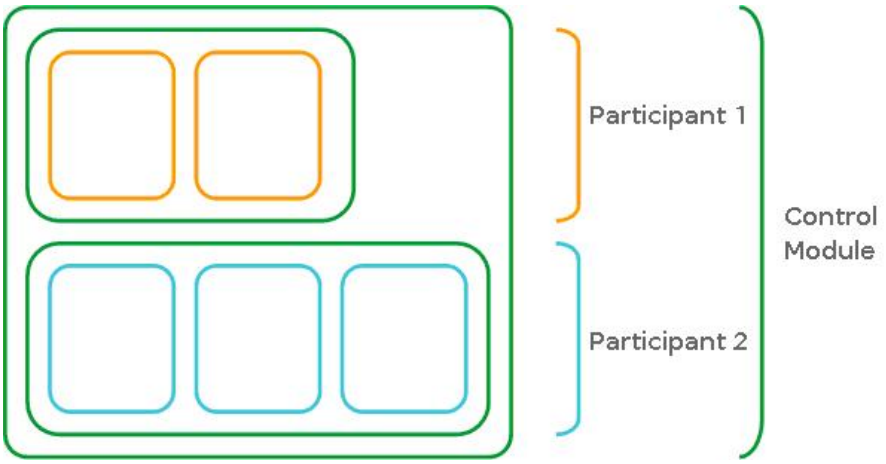
But, if the templates being designed could be used directly from the system application and linked by means of interface links, you need to provide the right connectivity between them (the user making the system application can only make interface links for establishing the relationship between different instances; whereas, the library designer has the opportunity to access the elements into the interfaces).

Composition Approach

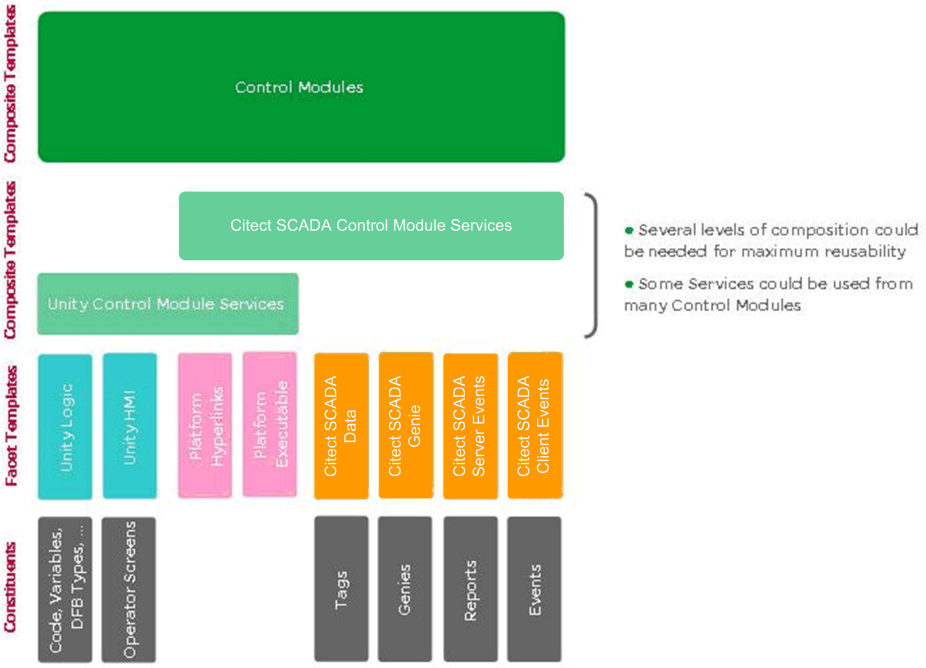
The objective is to provide composite templates which bring the features required for a control module (as defined in ISA-S88) that involves one or more participants when needed. Avoid the use of design composite templates with too many nested levels to avoid big changes propagation and to facilitate their maintainability. The libraries provide facet and composite templates which could be used to create instances or as part of other composite templates. This modular design allows the user to instantiate such templates in a consistent way for creating the application, independent of the participants involved and the required services (optional elements in the composite templates).

The first level of encapsulation is by participant. So, a horizontal approach is implemented. Composite templates offer the functionality from a concrete participant that can be combined for providing the complete functionality expected for a type of control module (that is, Motor 1 speed 1 direction).

The following figure shows general composition approach.



The following figure shows resources for application templates.



Another aspect to be considered is the perimeter for the control module templates.

The following capabilities are encapsulated by means of independent control module templates that the user can link in instantiation time:

- Foundation templates: Hardware Projection, Hardware Diagnosis, Peer to Peer, and so on.
- Communication templates.
- Devices SA templates.
- Process SA templates: Optionally the process templates include the projection of their related hardwired signals to facilitate the process of creation of hardwired signals with a minimum effort.

Exposed Interfaces

The templates expose interfaces that allow the user to make the following types of connections:

- To topological instances (mapping interfaces): When the application template requires some data provided by the topological instances (that is, I/O addresses), they are exposed through the mapping interfaces. Such types of interfaces can be used only from facet templates (not from composite templates) as the mapping process is executed from facets assigned to projects.
- To other application instances (application interfaces): The templates offer at least the capability to make the more usual connections between them (that is, Analog Input -> PID -> Analog Output) at the system level when such connections are complex (several data to be shared). This means that the user could create some logical connections between instances created directly at the system application.

  The standard library does not offer low level services which enable the user to do the following:

  - Make any connection between application instances.
  - Replace the services offered by the participants themselves (that is, FBD capabilities to define control logics for implementing Interlocks).

The interface names help the user to understand the concrete meaning of the interfaces in their specific scope.

So, even when the interface definition name and the chosen role are determining the data to be exchanged, the name of the interface allows the user to do the following:

- to determine its specific purpose.
- to distinguish between interfaces exposed by the same template which were created from the same interface definition.

Naming Convention

General

A key point for harmonization of the templates is to use a standard naming convention.

The general rules are applied depending on the following type of template being defined:

- Any types of templates
- Any interface models
- Mapping interface models
- Application interface models
- Application facet templates
- Application composite templates

Any Types of Templates

The following rules are applied to any types of templates:

- Prefix: The character $ is used as a prefix for templates made by Schneider Electric (Standard templates including interface models). This is avoiding any overlap with templates created by other users.
- Aliases: The objects model is bringing the possibility to use aliases to hide the origin of the deferred selections, parameters and interfaces. In the standard library, we are not using such capability. So the user creating references (Definition time) or instances (Instantiation time) will see the real name of the related selections, parameters, and interfaces.

- Documentation: Templates need to include references for enabling the access to their related engineering documentation (that is, user manuals). Such documentation needs to be attached to the related function (that is, `MotorGP` explanation as part of the composite where the Control Logic and the Control HMI facets are encapsulated).

  Even when such facets are not explicitly in some of the drawings included in this document, they are considered implicitly as included.

- Constituent names: To allow using multiple instances of the same template into the same participant project, it is commonly required to generate unique identifiers. The strategy to be applied is to add suffixes as in EcoStruxure Process Expert (for instance, `M1001_MotorGP`, `M1001_MotorGP_ST`, `M1001_CONDSUMGP`, and so on). That is also confirming the compatibility with previously created resources (that is, Smart Genies for representing the motor implemented through the Control function `MotorGP`).

  NOTE: You can use up to 18 characters while naming.

  Verify that the templates are establishing similar suffixes for avoiding the names conflicts. The designed templates force the user to use the same naming convention when creating new composite templates. The designed templates are created by recombining the ones coming from the standard libraries. This process increases their reusability even for the standard templates.

  So, the constituents names are calculated by means of bindings by the following rules:

  - The root of the constituent names is generated by using the name of the instance entered by the user.
  - The suffixes need to be added only when the fact of not using them is creating a name conflict at the template level that is being defined. This means that establishing suffixes need to be delayed as much as possible to the top composition levels.
  - The suffixes should follow the naming convention applied in EcoStruxure Process Expert and need to be useful for understanding the role of the constituents in the concrete scenario.

  For instance:

  - The user creates an instance of the composite template `$MotorGP` and enters the name `M1001` for it. This is reducing (but not avoiding) names conflicts (in fact, they can be only avoided by the participant in generation time regardless the chosen naming convention).

    NOTE: You can use up to 18 characters while naming.

  - This name is propagated to encapsulated templates through bindings without any suffices up to the level where that is causing a conflict (see the example in the next point).
  - When combining the Control functionality for controlling the motor (`MotorGP`) and the detected failures management (`CONDSUMGP`), it is required to concatenate the related suffixes (`M1001_MotorGP` and `M1001_CONDSUMGP`) to avoid creation of 2 DFBs with the same name, which is not allowed in Control projects.
  - When the facet defines the required constituents for implementing the control part of the object (`MotorGP`), it is required to concatenate additional suffixes (`M1001_MotorGP_ST` for the status data, `M1001_MotorGP_CFG` for the configuration data) for avoiding to create duplicate names.
  - In such case, if the user decides to instantiate directly the last mentioned facet template, it provides that no generated objects have duplicate names (into the scope of the instance). The template does not force the user to use the complete naming convention. For instance, `M1001` (`MotorGP`), `M1001_ST` and `M1001_CFG`; rather than `M1001_MotorGP`, `M1001_MotorGP_ST` and `M1001_MotorGP_CFG` generated when instantiating the template `$MotorGP`.

Any Interface Models

The following rules are applied:

- Roles representation:
  - Unidirectional interfaces: Female representation is used for the role in charge of producing the values for elements whereas Male is used for the other role.

Mapping Interface Models

The following rules are applied:

- Role names:
  - CO: Is used from the application templates representing communications objects (CO) that receive the data provided from the system configuration. This role is represented graphically as male.
  - DO: Is used from the application templates representing device objects (DO) that receive the data provided from the system configuration. This role is represented graphically as male.
  - HO: Is used from the topological template (also called Hardware Objects (HO)) to expose the information provided by the hardware (for example, topological addresses, slave number, and so on) to the project facets. This role is represented graphically as female.
  - SO: Is used from the application templates representing the signal objects SO) or I/O module objects that receive the data provided from the system configuration. This role is represented graphically as male.

Application Interface Models

As the same interface models can be used for several purposes and the objective of this key mechanism is to implement the collaboration between participants, their names need to be as generic as possible.

The naming convention is conditioned by the nature of the interface models.

The following types of interface models are considered for the standard library:

- Elementary interface models: The more elementary interface models are for exchanging just one element of an elementary type (not a nested Interface). These interfaces are for a generic usage as the interface name (the unique identifier) determines the type of the transported data.
- Single element interface models: Interfaces that contain just one element for a specific usage are named as the constituent type from the participant.
- Multiple elements interface models: Interfaces that contain several elements with a common objective. Instead of having several elementary interface models, one new interface model is created with many elements as needed.

The following rules are applied depending on the types:

- Any types of interface models
- Elementary interface models
- Single element interface models
- Multiple element interface models

Any Types of Interface Models

The following general rules are applied to any interface models:

- Interface model name: It is recommended to use upper and lowercase for better readability. The exception is when we use names that are directly used by the participant. In such case, the exact name is used.
- Role names:

  Definition and references: For example, a template is used for defining some constituents (that is, definition of Control variables) that could be used by one or more similar participants which need to access to such constituents (that is, references to Control variables).

  In such case, the role names to be used are:

  - Definition: Is for the role in charge of defining the constituents. This role is represented graphically as female.
  - Reference: Is for the role in charge of accessing (it does not matter for which kind of access: read, write, read/write, and so on) to such constituents. This role is represented graphically as male.

Elementary Interface Models

- Interface model name: The name of the real-time data type being exchanged is directly used for identifying the interface model (that is, $Bool for exchanging the name of a boolean constituent). It is required to limit the usage of the data being exchanged.

  For instance, to a concrete participant, they could be added some suffixes for avoiding. Such interface models could be used outside of its expected scope.
- Element name: The element is identified as name.

Single Element Interface Models

- Interface model name: The name needs to be identical to the type of the data being exchanged as defined in the participant (that is, `$MotorGP` for exchanging the name of a Control variable of type `$MotorGP`).
- Element name: The element is identified as name.

Multiple Element Interface Models

- Interface definition name: The name should reflect the meaning of the set of elements (that is, **$MotorDataGP** for exchanging the names of the OPC items in the Control Logic which are needed for accessing from Supervision).

Application Facet Templates

As much as possible, the name of the facet templates needs to be closer to the name of the encapsulated constituents or elements. As there could be several facet templates addressing the same functionality but from different points of view (for instance, different facet types) of the participant, consider the possibility to add some suffixes for distinguishing them.

- Suffixes: The following suffixes are used for identifying the templates:
  - `_UL` for Control logic facet templates.
  - `_UH` for Control HMI facet templates.
  - `_CD` for Citect SCADA data facet templates.
  - `_CG` for Citect SCADA genie facet templates.
  - `_CR` for Citect SCADA server event facet templates.
  - `_CC` for Citect SCADA client event facet templates.
  - `_PH` for platform Hyper link facet templates.

Application Composite Templates

As the composite templates are used to combine functionality provided by other facet or composite templates, it should reflect the complete functionality being encapsulated.

Basically, there are 2 use cases:

- Control module: Typically the composite template combines functionality from several participants for representing the complete functionality required for a type of control module (that is, motor 1 speed 1 direction). In such case, the type of control module is abbreviated to determine the name of the template. It is recommended to use upper and lowercase for maximizing readability (that is, `$MotorGP`).
- Control module services: The contained functionality is provided by the same participant for covering some services required for implementing one or more types of control modules. In such case, a suffix is added for expressing their scope (that is, the suffix `_UC` for the template `$MotorGP_UC` is expressing that it encapsulates functionality required from the perspective of the Control for implementing a motor 1 speed 1 direction).

  Suffixes: The following suffixes are used for identifying the templates:
  - `_UC` for Control logic composite templates.
  - `_UH` for Control HMI composite templates.
  - `_CS` for Citect SCADA data composite templates.
  - `_CG` for Citect SCADA genie composite templates.
  - `_CR` for Citect SCADA server event composite templates.
  - `_CC` for Citect SCADA client event composite templates.
  - `_CS` for Citect SCADA (data and genie) composite templates.

Common Parameters

Asset and Display

The table describes the asset and display parameters of supervision:

Section	Parameter description	Type	Default value	Additional remarks
Asset	**Asset Priority**	`Enum`	*Normal (4)*	Priority of the asset on a scale of 1 to 5 and 9 where 1 is the highest priority.
	Physical location	`String`	–	Asset location and properties have no specific impact on the library operation as delivered but are instead available for user or system data.
	Custom 1 (alarm filter)		–	User defined properties for the asset as well as each of its alarms. The asset filters allow the asset type to be

Section	Parameter description	Type	Default value	Additional remarks
	Custom 2 (alarm filter)			used to enable rapid filtering of alarm and event data.
	Custom 3 (alarm filter)			
	Custom 4 (user defined)		–	User defined Properties for the asset.
	Custom 5 (user defined)			
	Custom 6 (user defined)			
Display	**Process page name**	String	–	Process graphic page name where equipment genie is available.
	Other page names (optional list)			Overview graphic page associated with the equipment.

Inputs (Digital Input)

The table describes the digital input:

Section	Parameter description	Type	Default value	Additional remarks
Configuration	**Negate signal**	Boolean	*FALSE*	True = Negates the input signal.
	Custom signal name	String	–	Name used for the generated EFB/DFB and variables.

Outputs (Digital Output)

The table describes the digital output:

Section	Parameter description	Type	Default value	Additional remarks
Configuration	**Negate signal**	Boolean	*FALSE*	True = Negates the output signal.
	Custom signal name	String	–	Name used for the generated EFB/DFB and variables.

Local Panel

The table describes the local panel that are applicable to the selected template:

Section	Parameter description	Type	Default value	Additional remarks
Historize	**Historize status/ commands**	Boolean	*FALSE*	It allows to historize the change in status or commands.
	Enable historize status/commands			Enable historize status or commands.
	Enable historize configuration word			Enable historize configuration word

Local Panel (Logic)

The table describes the local panel that are applicable to the selected template:

Section	Parameter description	Type	Default value	Additional remarks
Configuration	**Select local panel from faceplate**	`Boolean`	*FALSE*	–
	Disable setpoint from faceplate	`Enum`	*FALSE*	Allows disabling the setpoint command from faceplate when equipment is in local mode • **False Value**: Value “0” is assigned to the input pin `LockSetpointEn`. This condition allows to change the setpoint from faceplate if the selected **Owner** is **Local Panel**. • **True Value**: Value “1” is assigned to the input pin `LockSetpointEn`. This condition will not allow to change the setpoint from faceplate even if the selected **Owner** is **Local Panel**. • **Interface**: No value is assigned to the input pin `LockSetpointEn`. This condition allows to change the setpoint from faceplate by changing the state from digital input signal if the selected **Owner** is **Local Panel**. ◦ If digital input signal value “1” is passed then the setpoint cannot be changed from the faceplate. ◦ If digital input signal value “0” is passed then the setpoint can be changed from the faceplate. • **Refinement**: The link `DisSPSelection` will be deleted from the interface, hence allowing to create the logic from the refinement and connecting with the `LockSetpointEn`.
	Setpoint when pushbutton is pressed	`Float`	*2.0*	–
	Setpoint when pushbutton are held down	`Float`	*1.0*	–
	Maximum time before system interprets that command has repeated	`Duration`	*00:00:02*	For control valve

Alarm

The table describes the alarm parameters that are applicable to the selected template:

Section	Parameter description	Type	Default value	Additional remarks
Alarm	**Confirmation failure description** **Confirmation motor failure description** **Confirmation valve failure description**	String	*@(Confirmation failure)*	A detail description for alarms.
	Motor confirmation failure description		*@(Motor confirmation failure)*	
	Position confirmation failure description		*@(Position mismatch)*	
	Valve confirmation failure description		*@(Valve confirmation failure)*	
	Failure description (common mode) **Device failure description** **Device failure description (common mode)**	String	*@ (device failure)*	A detail description for alarms.
	Channel failure description	String	*@(Channel failure alarm)*	A detail description for alarms.
	Interlock trip description	String	*@(Interlock trip)*	A detail description for Interlock trip.
	Open Confirmation failure description	String	*@ (Open Confirmation Failure)*	-
	Close Confirmation failure description	String	*@ (Close Confirmation Failure)*	-
Alarm	**Confirmation failure severity** **Failure severity** **Confirmation motor failure severity** **Confirmation valve failure severity** **Device failure severity** **Motor confirmation failure severity** **Position confirmation failure severity** **Valve confirmation failure severity** **Channel failure severity** **Open Confirmation failure severity** **Close Confirmation failure severity**	Enum	*Low (4000)*	It allows to set the criticality of alarm.
	Interlock trip severity	Enum	*Low (4000)*	It allows to set the criticality of alarm.
Alarm	**Confirmation failure Group** **Failure group** **Confirmation motor failure group** **Confirmation valve failure group** **Interlock trip group** **Device failure group** **Motor confirmation failure group**	String	*Failure*	Group name for rapid alarm filtering.

Section	Parameter description	Type	Default value	Additional remarks
	Position confirmation failure group **Valve confirmation failure group** **Channel failure description** **Open Confirmation failure group** **Close Confirmation failure group**			
	Channel failure group	`String`	*(Channel)*	Group name for rapid alarm filtering.
Alarm	**Alarm failure delay**	`Duration`	*00:00:00*	–
	Failure disable	`Boolean`	*FALSE*	–
Alarm	**High high alarm**	`Enum`	*True Value (1)*	–
	High alarm			–
	Low alarm			–
	Low low alarm			–
	Deviation alarm			–
	Deviation setpoint calculation	`Enum`	*Absolute Value (0)*	–
	Deviation limit setpoint type			–

Trend

The table describes the local panel that are applicable to the selected template:

Section	Parameter description	Type	Default value	Additional remarks
Trend	**Sample period**	`Duration`	*00:00:05*	–
	History length (weeks)	`Integer`	*5*	–
	History rollover time	`Duration`	*12:00:00*	–
	History rollover day	`String`	*Tuesday*	–
	Storage type	`String`	*TRN_EVENT*	–
	Data location (optional)	`String`	*[Data]:*	–
	Disable local setpoint value	`Boolean`	*FALSE*	–
	Disable output value			–

Common Control Module Template Interfaces

Common Interface Parameters

The table describes the control module template interfaces:

Interface identifier	Type/Role	Description	Example
Bool	*$Bool*	An input interface of boolean type variable.	Different types of variables can be connected to different objects.
INT	*$Integer*	An input interface of integer type variable.	
Real	*$Real*	An input interface of real type variable.	
Time	*$Time*	An input interface of time type variable.	
OpenPosSwitch/ ClosePosSwitch/	*$Bool*	Feedback input interface from other template.	A boolean type of interface can be connected as feedback to the motor.
FWDILCK/REVILCK/ ILCK/ILCKClose/ ILCKOpen/OpenILCK/ CloseILCK	*$Bool*	Interlock input interface from other template.	Output (MotorOP and MotorSP) of a motor can be connected to input (ILCK) of another Motor as an interlock condition.
DEV2S1D/Dev1S1D/ Dev1S2D	*$DEV2S1D/ $Dev1S1D/ $DEV1S2D*	Motor controller Interface from the device objects like motor starters.	A motor starter (TESYST) can be connected to control the motor.
RSPSEL/RSPREV	*$Bool*	Remote setpoint selection input interface from other template.This interface is used to determine the direction of motor when it is remotely controlled.	The forward output of a two direction motor can be connected as remote direction setpoint to RSPSEL.
RSP/RemoteSetpoint	*$Bool/Real*	Remote setpoint input interface from other template.	An output of a motor can be connected as a remote setpoint to other motor.
Rstart	*$Bool*	Remote start input interface from other template.	A remotely controlled motor can be controlled by an open output of a valve.
Rstop/RemoteStop	*$Bool*	Remote stop input interface from other template.	
INTVar	*$Integer*	Initial variable name input/Output Interface interface to analog input or output template.	An integer variable can be connected to analog input if AI signal is not connected.
Name	*$String*	Instance name output interface to other template.	Instance name of a valve can be used by other valve.
MotorSP/Motor2SP1/ Motor2SP2/ValveSP/ OpenSP/CloseSP/ MotorOP/Motor2OP/ Motor2OP1/ Motor2OP2/ValveOP/ OpenOP/CloseOP/ MotorLOP/Motor2LOP/ Motor2LOP1/ Motor2LOP2/ ValveLOP/OpenOPL/ CloseOPL	*$Bool*	Setpoint, output or logical output interface to other template.	The setpoint, output or logical output of a motor can be used as a remote setpoint of other remotely controlled motor.
OpenPos/ClosePos/ Running Forward/ Running Reverse	*$Bool*	Current feedback output interface to other template.	Current feedback of a motor can be connected as setpoint for other valve.
PV	*$Bool/$Real/ $Integer*	Current Value in measurement is an input or output interface to and from template.	The measured speed from the variable speed drive can be connected to MotorVS as an input present value. The Current Value in engineering unit from an analog input can be connected to a PID block.
PVRange/SPRange/ PVRanged/SPRanged/ RSPRanged/OUTrange/ OUTRanged/AORange	*$RangeGP*	Input or output interface to define the range of a value.	A range block can be connected to analog input to specify the range of a MotorVS measured speed or a range output interface of analog input can also be connected to PID.
RSPRangedwTrack/ OUTRangedwTrack	*$RangedValue-withTracking*	Input/Output interface to an object by providing the range for a values along with track of output status.	A range block with track can be connected to control valve to specify the range and also to configure track variable to the output.
OPInc/OPDec	*$Bool/Def*	Logical output to increase/decrease present value.	–
DOUTPUTSignalExt	*$DOUTPUTSIg-nal*	Digital output signal external with quality.	–

Common Interface Parameters

The table describes the control module template interfaces:

Interface identifier	Type/Role	Description	Example
Bool	*$Bool*	An input interface of boolean type variable.	Different types of variables can be connected to different objects.
INT	*$Integer*	An input interface of integer type variable.	
Real	*$Real*	An input interface of real type variable.	
Time	*$Time*	An input interface of time type variable.	
OpenPosSwitch/ ClosePosSwitch/	*$Bool*	Feedback input interface from other template.	A boolean type of interface can be connected as feedback to the motor.
FWDILCK/REVILCK/ ILCK/ILCKClose/ ILCKOpen/OpenILCK/ CloseILCK	*$Bool*	Interlock input interface from other template.	Output (MotorOP and MotorSP) of a motor can be connected to input (ILCK) of another Motor as an interlock condition.
DEV2S1D/Dev1S1D/ Dev1S2D	*$DEV2S1D/ $Dev1S1D/ $DEV1S2D*	Motor controller Interface from the device objects like motor starters.	A motor starter (TESYST) can be connected to control the motor.
RSPSEL/RSPREV	*$Bool*	Remote setpoint selection input interface from other template.This interface is used to determine the direction of motor when it is remotely controlled.	The forward output of a two direction motor can be connected as remote direction setpoint to RSPSEL.
RSP/RemoteSetpoint	*$Bool/Real*	Remote setpoint input interface from other template.	An output of a motor can be connected as a remote setpoint to other motor.
Rstart	*$Bool*	Remote start input interface from other template.	A remotely controlled motor can be controlled by an open output of a valve.
Rstop/RemoteStop	*$Bool*	Remote stop input interface from other template.	
INTVar	*$Integer*	Initial variable name input/Output Interface interface to analog input or output template.	An integer variable can be connected to analog input if AI signal is not connected.
Name	*$String*	Instance name output interface to other template.	Instance name of a valve can be used by other valve.
MotorSP/Motor2SP1/ Motor2SP2/ValveSP/ OpenSP/CloseSP/ MotorOP/Motor2OP/ Motor2OP1/ Motor2OP2/ValveOP/ OpenOP/CloseOP/ MotorLOP/Motor2LOP/ Motor2LOP1/ Motor2LOP2/ ValveLOP/OpenOPL/ CloseOPL	*$Bool*	Stpoint, Output or logical output interface to other template.	The setpoint, output or logical output of a motor can be used as a remote setpoint of other remotely controlled motor.
OpenPos/ClosePos/ Running Forward/ Running Reverse	*$Bool*	Current feedback output interface to other template.	Current feedback of a motor can be connected as setpoint for other valve.
PV	*$Bool/$Real/ $Integer*	Current Value in measurement is an input or output interface to and from template.	The measured speed from the variable speed drive can be connected to MotorVS as an input present value. The Current Value in engineering unit from an analog input can be connected to a PID block.
PVRange/SPRange/ PVRanged/SPRanged/ RSPRanged/OUTrange/ OUTRanged/AORange	*$RangeGP*	Input or output interface to define the range of a value.	A range block can be connected to analog input to specify the range of a MotorVS measured speed or a range output interface of analog input can also be connected to PID.
RSPRangedwTrack/ OUTRangedwTrack	*$RangedValue-withTracking*	Input/Output interface to an object by providing the range for a values along with track of output status.	A range block with track can be connected to control valve to specify the range and also to configure track variable to the output.

Interface identifier	Type/Role	Description	Example
`OPInc/OPDec`	*$Bool/Def*	Logical output to increase/decrease present value.	–
`DOUTPUTSignalExt`	*$DOUTPUTSIg-nal*	Digital output signal external with quality.	–
`ExtendedData`	*ASelectINF/B*	Analog selector extended data interface.	ExtendedData interface is used to link Advanced PID and Analog Select

Hyperlink Services

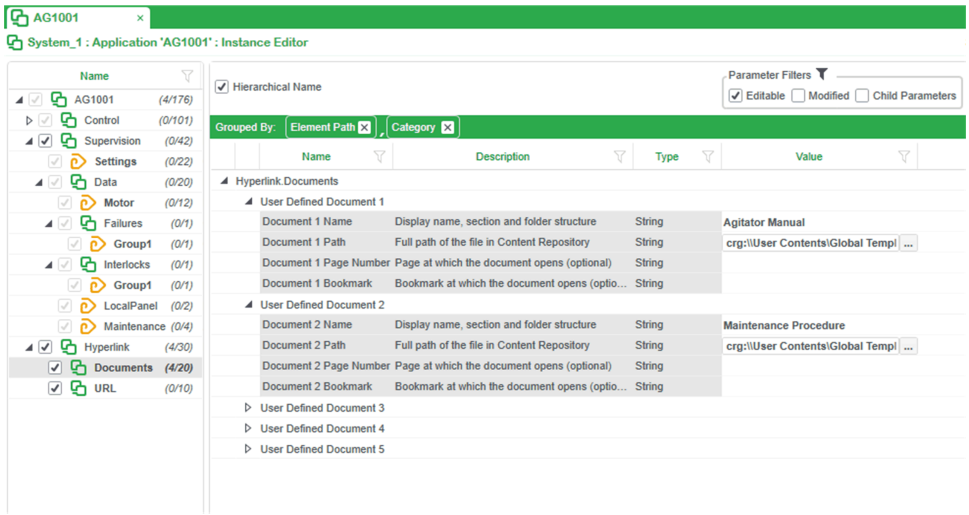
Hyperlink

EcoStruxure Process Expert provides the user links to access documents associated with an asset through the operation client (runtime navigation services). Hyperlinks are defined within the library to provide access to elements configured by the library.

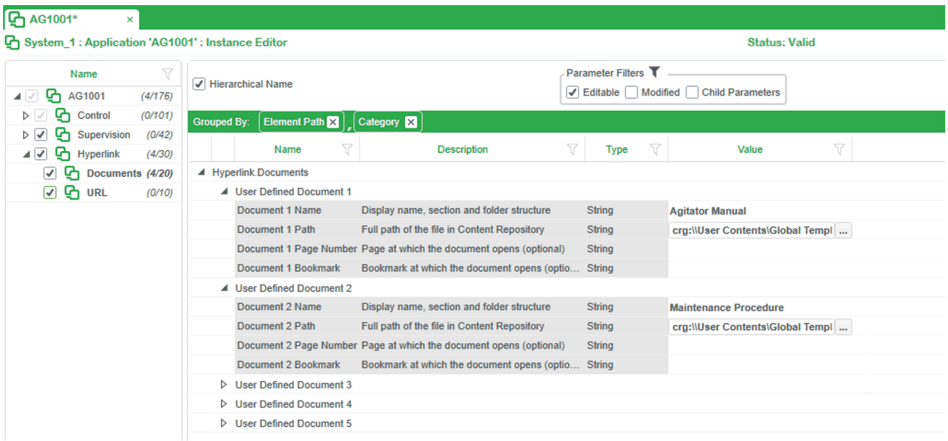
Hyperlinks are also available as configurable parameters for the user to provide their own links to the documents and web pages such as:

- Equipment Manuals
- Process and Instrumentation Diagrams
- Maintenance Procedures
- Maintenance Requests
- Reports

This figure shows the `HyperLink` configuration to the documentation and web pages:

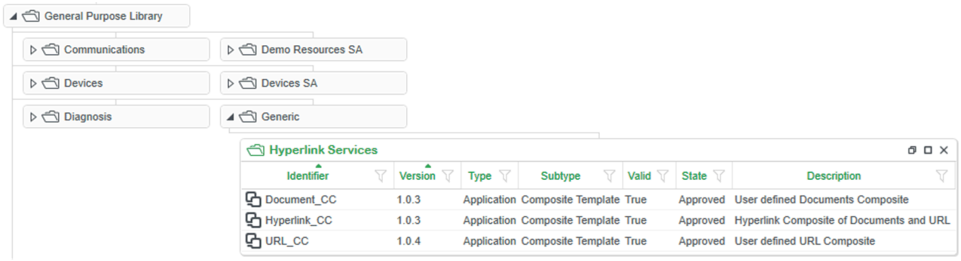


This figure shows the Runtime Navigation to the documentation and web pages:



Hyperlink composites are stored in *Hyperlink* Services folder under *Generic* folder:

This figure shows the Hyperlink services folder:



Control

What's in This Chapter

Modularity.....52
Function Block Interface 52

Overview

The EcoStruxure Process Expert software provides the resources that have been pre-configured and tested by Schneider Electric and that were specifically designed for automating a large variety of processes.

The control resources for process control provide the common required functions, facilitating the development of control systems.

To automate and simplify the implementation process of control systems, you can use these resources together with tools for code generation and for the synchronization of control and monitoring subsystems.

This document describes the basic concepts and details behind each one of the function blocks (DFBs) for implementing the common cross-process and cross-market EcoStruxure control functions.

Modularity

Overview

The function blocks are designed in such a way that the functions they implement are cumulative.

This way, you can incorporate the functions required for each specific case into the system.

Example

An example featuring three function blocks connected to each other to provide the necessary functions to set up an on-off motor with interlocking and maintenance. A `MotorGP` block (core service) provides on-off motor functionalities. Additionally, you can integrate `ILCKONGP/ILCKOFFGP` block (optional service) to achieve interlock functionalities and `DEVMNTGP` block (optional service) to provide maintenance functionalities (number of switch operations and hours of operation).

Function Block Interface

Overview

The function blocks for Process provide an interface that allows them to be configured, monitored, and controlled both from the monitoring subsystem and the control subsystem (continuous and/or sequential control).

The following interfaces are provided:

- Basic Configuration
- Continuous Control
- Sequential Control

- States and Monitoring

Basic Configuration

DFB input pins are usually connected to static data and recognized in engineering time (for example, input channel range or limit switch enabling on an on-off motor).

Continuous Control

DFB input and output pins:

- allow receiving commands from other blocks.
- provide block status to other blocks to enable implementing switching operations (for example, remote setpoint -`RSP`-), detected alarms (for example, high-level alarm -`HIHI`-), interlocking (for example, activation of interlocking `ILCK`, or interlocking setpoint -`ILCKPos`-), and so on.

Sequential Control

SC public and structured variable publishes the block status and allows its control from the control sequences (commands not kept).

States and Monitoring

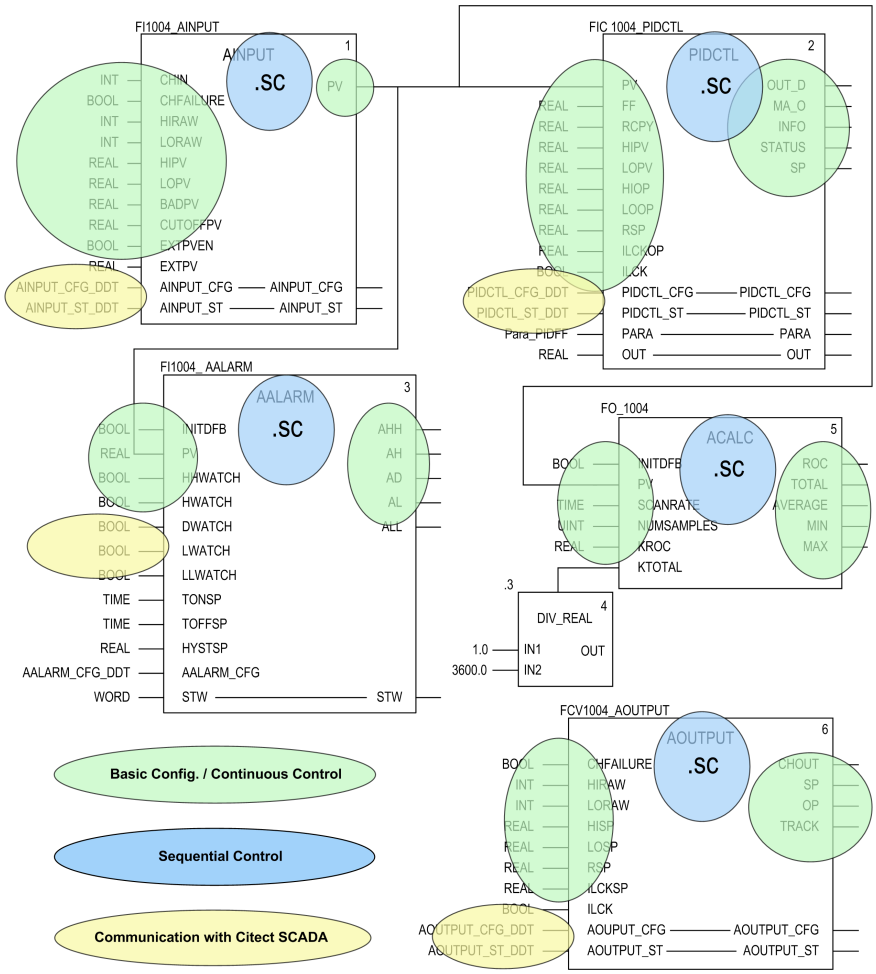
Depending on their types, the blocks feature up to 2 input/output pins that need to be connected to variables used to maintain the pin states of the blocks. In addition, these variables allow the commands and parameters received from the monitoring subsystem to be managed.

The frequently considered variables are:

- State (identified with the _ST suffix): These variables hold the state and control characteristics used from the first monitoring subsystem level (dynamic symbols in the flowcharts).
- Configuration (identified with the _CFG suffix): These variables hold second-level information for the parameter configuration of the block. This information is accessible from the monitoring subsystem faceplates.

Illustration

The following figure illustrates the previous provided interface.



Supervision

What's in This Chapter

Supervision Services 55
Access Control 56
Calculated Variable 58
Commonly Used Items 59
Genies 61
Faceplate 65
Color Codes 71

Overview

This part describes the supervision services.

Supervision Services

Supervision Resource Structure

Types of Resources

The GeneralPurposeLibrary project contains the following Supervision resources grouped in libraries:

Symbols	Static graphical elements are supplied as symbol libraries so that you can easily redefine the appearance of Supervision components.
Genies	Dynamic objects are supplied as Supervision genies (in turn, these can be made up of other genies) to represent the dynamic elements that are commonly used in monitoring flowcharts.
Faceplates	Genies are linked to faceplates that allow you to access parameters such as operation, configuration, or maintenance parameters, depending on the function. Faceplates are implemented by using Supervision pages featuring supergenie syntax (refer to the Supervision Participant help for details).
Templates	Template types with various resolutions featuring the standard EcoStruxure Process Expert format.
Cicode	The functions that allow genies and their faceplates to work properly are implemented by using the proprietary Supervision Participant programming language.

Deploying Supervision Resources

When you edit pages, EcoStruxure Process Expert adds the GeneralPurposeLibrary project to the Supervision Participant project so that the resources it contains are deployed with the master project.

Access Control

Privilege Levels

Overview

Supervision components feature access control so that only users having access to the configured area and the required privilege level execute actions on dynamic objects and/or faceplate during operation.

You need to assign privilege levels and areas to users to configure their access permissions depending on the specific requirements of each application.

An Area is configured for each instance. Based on the roles, privilege is assigned to the area. User assigned with the privilege to an area within a role has the view access to that area. However, the user can only operate the system elements in that area having appropriate privilege. For more details, refer to Plant SCADA help.

The privilege levels are mentioned in the custom *8* fields of the item. It is either a single digit or a two digit number. If it is a single digit number, then the privilege is equivalent to the Citect philosophy, refer to Citect SCADA help.

The double-digit privilege level is applicable for write command confirmation.

Default Privilege Levels

By default, the following general criteria are defined to assign privilege levels in Supervision components:

Privilege	Role
0	Everyone
1	Operator
2	Supervisor
3	Maintenance
4	Not used
5	Engineer
6	Not used
7	Administrator
8	Not used

Confirmation

Confirmation operation is applicable to write the command. Write operation is confirmed by adding the second digit to the privilege.

Following are two types of confirmation based on the privilege number:

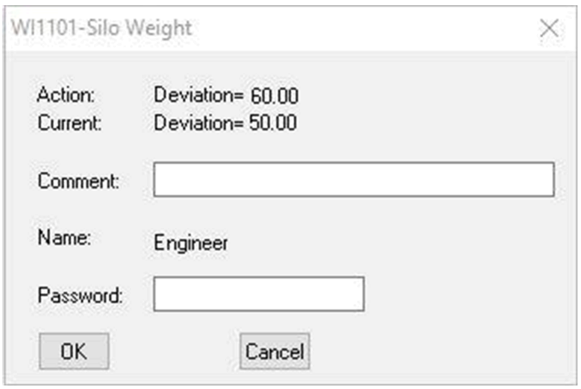
- Single confirmation
- Double confirmation

Single Confirmation

Confirmation from the currently logged-in user is required. The user has to perform the confirmation operation if the second digit of privilege level is *0*. For example, if the privilege level is set to *50* for the item, then the engineer confirmation *(5 + 0)* is required to perform an operation. This is referred to as

`Engineer (Confirmed)`, `Operator (Confirmed)`, and so on, based on the privilege level.

Example:

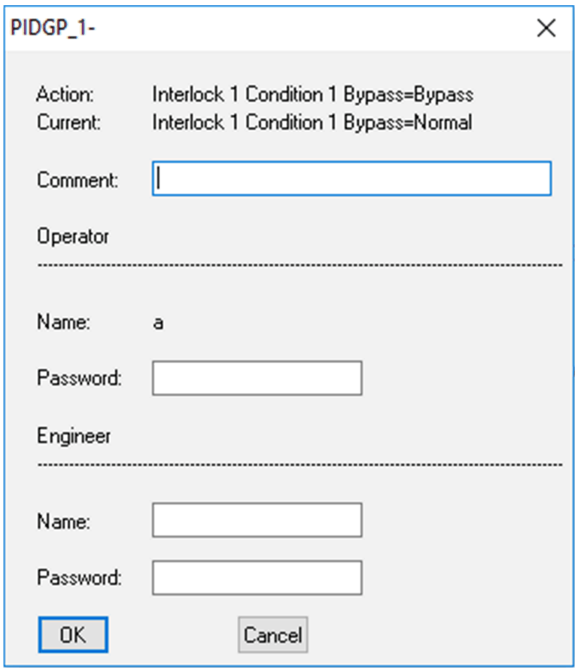


In earlier example, a login user is `Engineer`. when the `Engineer` changes the deviation limit from *50 to 60* of silo weight, a security popup appears and the `Engineer` must enter the password again to confirm the operation.

Double Confirmation

Confirmation from currently logged-in user and verifier user who has privilege equal to the second digit is required. The user has to perform the confirmation and verify the operation if the second digit of privilege level is not *0*. If the privilege level is set to *15* for the item, then the `Operator` and `Engineer` *(1 + 5)* are required to perform operation. This is termed as `Operator` + `Engineer`, `Operator` + `Supervisor`, and so on, based on the privilege level.

Example:



In earlier example, the user is `Operator`. When the `Operator` changes the deviation limit from *50 to 60* of silo weight, a security popup appears and the `Operator` enters the password again to confirm the operation and `Verifier` needs to enter the password to verify the operation.

Calculated Variable

Calculated Variable

Overview

A calculated variable allows you to generate an item value at runtime that is the result of a Cicode address and call an internal Cicode function as a variable item.

Operating Owner

Following are the operating owner:

- **Operator**: Control module is operated by the supervision operator.
- **Cascade**: Control module is operated by continuous control.
- **Program**: Control module is operated by the sequential control/equipment module/InBatch.
- **External**: Control module is controlled from outside of the system. Monitoring of confirmation alarms is stopped during external control. The detected error alarm has to be considered.
- **Local panel**: Control module is operated by the operator in the field. If the setpoint is not locked in control, then it has to be operated from supervision.
- **Zero**: Selection of zero owner in the local panel isolates the control module. You cannot operate the control module from the supervision and the field (local panel).

Owner symbol shows the abnormal operating owner on the genie. Owner symbol displays, if the current owner is abnormal.

The table describes the current owner calculation:

Owner Enumeration	Owner Symbol	Owner Value	External Owner = 1	Program Owner = 1	Cascade Owner = 1	Operator Owner = 1	Zero Owner = 1	Local Panel Owner = 1
Unknown	**–**	*0*	No	No	No	No	No	No
Operator	**O**	*1*	No	–	–	Yes	–	–
Program	**P**	*2*	No	Yes	Yes	–	–	–
Cascade	**C**	*3*	No	–	–	–	–	–
Local Panel	**LP**	*4*	No	–	–	–	–	Yes
Zero	**Z**	*5*	No	–	–	–	Yes	–
External (O)	**E**	*11*	Yes	–	–	Yes	–	–
External (P)	**E**	*12*	Yes	Yes	–	–	–	–
External (C)	**E**	*13*	Yes	–	Yes	–	–	–
External (LP)	**E**	*14*	Yes	–	–	–	–	Yes
External (Z)	**E**	*15*	Yes	–	–	–	Yes	–

Owner Function

The table describes the Cicode function to be used based on the available owner (represented by x) in the control module:

Function	Pro-gram	Opera-tor	Cas-cade	Local Panel	Zero Mode	Exter-nal
GPL_OwnerExternalLocal	X	X	X	X	X	X
GPL_OwnerExternal	X	X	X	–	–	X

Function	Program	Operator	Cascade	Local Panel	Zero Mode	External
GPL_OwnerBasic	X	X	–	–	–	–
GPL_Owner	X	X	X	–	–	–
GPL_OwnerLocal	X	X	X	X	X	–
NOTE: This function is generated based on the instance configuration level.						

Operating Mode

Following are the modes of operation:

- **Normal**: Allows the control module to be used as part of the control system.
- **Override**: Allows the control module to continuously operate in the event of a detected feedback error. In this mode, input feedback is overridden.
- **Simulation**: Allows the logic of sequences or control modules to be tested. The inputs to the control system are overridden and the outputs suppressed. The control modules operate in a simulated environment.
- **Out of Service**: Excludes the control module from operation within the control system.
- **Maintenance**: Allows the free operation of the control module to maintain and test the control module. The inputs and outputs are active but the operation of the control module is hidden from the automation logic. Also, interlock and detected error conditions are bypassed.

Mode Function

The table describes the Cicode function to be used based on the applicable modes (represented by x) in the control module:

Function	Normal	Override	Out of service	Simulation	Maintenance
GPL_Mode	X	X	X	X	X
GPL_ModeSignalInput	–	X	–	–	X
GPL_ModeSignalOutput	–	–	X	X	X
GPL_ModeMValve	–	–	X	X	X
GPL_ModeBasic	X	X	–	–	–

Commonly Used Items

Commonly Used Items

Maintenance Variable Items

The table describes the variable items that are used by Supervision components for *maintenance*:

Item Name	Description	Enumeration	Address
Bit items derived from MntStatusWord.			
`DevMntHoursOverflow`	Hours count overflow status.	On/Off	`Bit0`
`DevMntOperationsOverflow`	Operations count overflow status.		`Bit1`

Item Name	Description	Enumeration	Address
Bit items derived from MntConfigurationWord.			
DevMntResetHoursCounter	Reset the hours counter.		Bit0
DevMntResetOperationsCounter	Reset Operations Counter.		Bit1
IO items			
DevMntMntStatusWord	Maintenance status word	–	DEVMNTGP_STW
DevMntConfigurationWord	Maintenance configuration word		DEVMNTGP_CFGW
DevMntOperationHours	Total run hours	–	DEVMNTGP_CFG_TotalRunHours
DevMntManeuversNumbers	Total starts		DEVMNTGP_CFG_TotalStarts

Interlock Variable Items

The table describes the variable items that are used by Supervision components for *Interlock*:

Item Name	Description	Address
ILock{x}Condition{y}	Condition {y} status of interlock group{x}.	ILock{x}ConditionWord.Bit{y-1}
NOTE: {x} represents the group, range from 1 to 4. {y} represents the condition, range from 1 to 7.		
ILock{x}ConditionResult	Result of the conditions of interlock group{x}.	ILock{x}ConditionWord.Bit15
NOTE: {x} represents the group, range from 1 to 4.		
ILock{x}BypassCondition{y}	Condition {y} bypass status of interlock group {x}.	ILock{x}ConditionWord.Bit{y-1}
NOTE: {x} represents the group, range from 1 to 4. {y} represents the condition, range from 1 to 7.		
ILock{x}RearmCondition{y}	Condition {y} rearm required status of interlock group{x}.	ILock{x}RearmWord.Bit{y-1}
NOTE: {x} represents the group, range from 1 to 4. {y} represents the condition, range from 1 to 7.		
ILock{x}Disable	Indicates that interlock group{x} is disabled.	ILock{x}BypassWord.Bit15
NOTE: {x} represents the group, range from 1 to 4.		
ILock{x}FirstCondition{y}	Indicates that condition {y} is the first condition which interlocked the equipment.	ILock{x}FirstWord.Bit{y-1}
NOTE: {x} represents the group, range from 1 to 4. {y} represents the condition, range from 1 to 7.		
IO items		
ILock{x}ConditionWord	Interlock condition word	ILCKGP_ST.COND
ILock{x}BypassWord	Interlock bypass word	ILCKGP_ST.BYPASS
ILock{x}RearmWord	Interlock rearm word	ILCKGP_ST.REARM

Item Name	Description	Address
ILock{x}FirstWord	Interlock first word	ILCKGP_ST.FIRST
NOTE: {x} represents the group, range from 1 to 4.		

Failure Variable Items

The table describes the variable items that are used by Supervision components:

Item Name	Description	Source
Fail{x}Condition{y}	Condition {y} status of detected error group{x}.	Fail{x}ConditionWord.Bit{y-1}
NOTE: {x} represents the group, range from 1 to 4. {y} represents the condition, range from 1 to 15.		
Fail{x}ConditionResult	Result of the conditions of detected error group{x}.	Fail{x}ConditionWord.Bit15
NOTE: {x} represents the group, range from 1 to 4.		
Fail{x}BypassCondition{y}	Condition {y} bypass status of detected error group{x}.	Fail{x}ConditionWord.Bit{y-1}
NOTE: {x} represents the group, range from 1 to 4. {y} represents the condition, range from 1 to 15.		
Fail{x}RearmCondition{y}	Condition {y} rearm required status of detected error group {x}.	Fail{x}RearmWord.Bit{y-1}
NOTE: {x} represents the group, range from 1 to 4. {y} represents the condition, range from 1 to 15.		
Fail{x}Disable	Indicates that detected error group{x} is disabled.	Fail{x}BypassWord.Bit15
NOTE: {x} represents the group range from 1 to 4.		
Fail{x}FirstCondition{y}	Indicates that condition {y} is the first condition which made equipment inoperable.	Fail{x}FirstWord.Bit{y-1}
NOTE: {x} represents the group, range from 1 to 4. {y} represents the condition, range from 1 to 15.		
IO items		
Fail{x}ConditionWord	Detected error condition word	CONDSUMGP_ST.COND
Fail{x}BypassWord	Detected error bypass word	CONDSUMGP_ST.BYPASS
Fail{x}RearmWord	Detected error rearm word	CONDSUMGP_ST.REARM
Fail{x}FirstWord	Detected error first word	CONDSUMGP_ST.FIRST
NOTE: {x} represents the group, range from 1 to 4.		

Genies

Overview

This section provides general information about the representation of genies that are part of the GeneralPurposeLibrary project.

Using Genies

Genie Availability

The GeneralPurposeLibrary resource contains a number of genies grouped in libraries that correspond to Supervision functions. For example, the `gpl_motor` library contains nine genies for on/off motor management.

These genies allow monitoring and interacting with the control modules during operation.

Genies become visible in runtime once you assign them to a Supervision page.

Assigning Genies

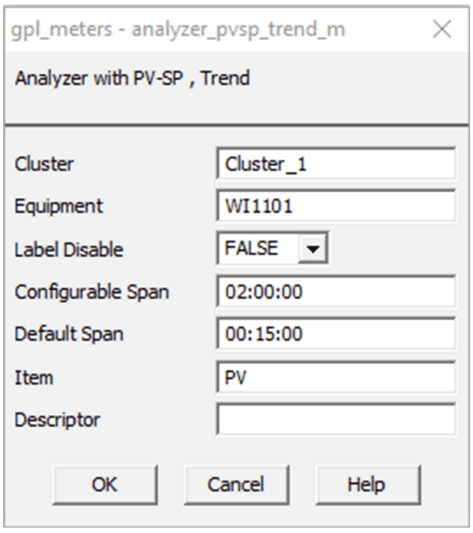
You can assign the genies that are referenced to a Supervision page by editing the page, using the **Edit** command.

Genie Properties

Once you have assigned a genie to a page, double-click it to display its properties.

Genies of the GeneralPurposeLibrary project have the **Cluster** and **Equipment**, **Label Disable**, **Configurable Span**, **Default Span**, **Item**, and **Descriptor**, properties.

The figure shows an example of the properties dialog box of the `analyzer_PVSP_trend` genie:



The table describes the properties of the `analyzer_PVSP_trend` genie:

Genie properties	Description
Cluster	Cluster name of the equipment.
Equipment	Equipment name
Label Disable	By default, Label Disable is False, enter True if you need to hide the label of an object.
Configurable Span	Displays the value that was selected last by using the span selection button of the genie.
Default Span	Displays default time span for the trend view in HH:MM:SS.
Item	Item name of the equipment which has to be plotted on the trend.

Genie properties	Description
	NOTE: If item is not specified, the pens will be added for eight items (PV, SP, LowLowLimit, LowLimit, HighLimit, HighHighLimit, ORHigh, and ORlow) of the equipment.
Descriptor	Descriptor entry appears on the genie property dialog box. The descriptor is applicable for the bar horizontal, bar vertical, deviation, and analyzer genies of the `gpl_meters library`.
NOTE: Item, Configurable Span, Default Span, and Descriptor parameters are applicable only for analog inputs.	

Genie Icons

Description

The table describes the icons that are displayed during an operation:

Function	Icon	Meaning	Examples of use	Comment
Displaying owner selection	**P**	Program	Valve in Program Owner mode (the program sets the setpoint).	The icon is displayed only if it is not the normal mode.
	C	Cascade	Closed-loop PID controller with an externally set setpoint	–
	O	Operator	Valve in Operator Owner mode (the operator sets the setpoint).	–
	E	External	–	–
	L	Local	Valve is operated locally from the local panel	–
	Z	Zero	Valve is in Idle mode not controlled from control system or local panel	–
Displaying Equipment state information		Active State	Motor is in running state.	This icon only be visible when The `EquipState` Hide is created in the SCADA parameters. To create, go to Setup > Parameters. Enter the `GPL.EquipState` in Section Name, Hide in Name and set the Value = 1.
		Passive State	Motor is in stopped state	
Interlocking		Active	A motor is interlocked because of an interlock condition.	The Icon will not be visible for sequential interlock when asset in configured under group control and `EquipState`hide parameter is created in the SCADA.
Displaying service information		Out of service	A motor is declared out of service in its faceplate.	–
Displaying maintenance mode		Maintenance	A motor is declared in maintenance mode	–
Transitioning		Transitioning state	Valve opening	Active transitioning
			Valve closing	Passive transitioning
Informing of abnormal configuration		The control module is in abnormal configuration.	The control module is in simulation.	Represented on the element that is in abnormal configuration.
Requesting operator action		The control module is waiting for an operator action.	A motor needs a reset to start again.	Flashing of the icon if any action required

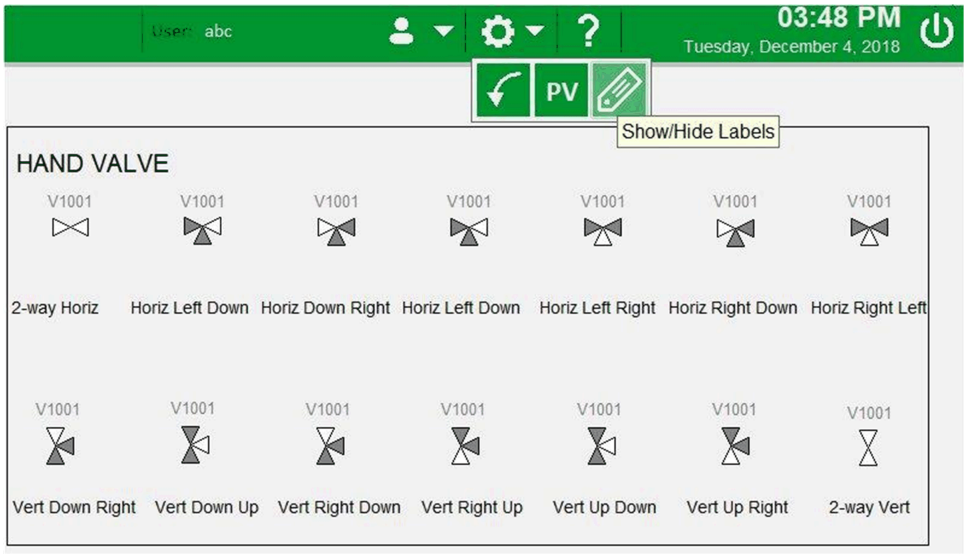
Function	Icon	Meaning	Examples of use	Comment
Error icon		Detected error status of the equipment.	When motor is inoperable.	The icon is displayed when the motor is detected error state.
Displaying alarms		Alarm of priority *1*	Very high temperature alarm.	*Critical alarm ON unacknowledge*
		Alarm of priority *1* returns to normal	–	*Critical alarm OFF unacknowledge*
		Alarm of priority *2*	High temperature alarm	*AlarmBorder high unacknowledge*
		Alarm of priority *2* returns to normal	–	*AlarmBorder high unacknowledge*
		Alarm of priority *3*	Function check (Namur status).	*AlarmBorder medium unacknowledge*
		Alarm of priority *3* returns to normal	–	*AlarmBorder medium unacknowledge*
		Alarm of priority *4*	–	*AlarmBorder low unacknowledge*
		Alarm of priority *4* returns to normal	–	*AlarmBorder low unacknowledge*
Indicating alarm level setpoints		Very high	Very high temperature	–
		High	High temperature	–
		Setpoint	Temperature outside of setpoint	–
		Deviation	Temperature outside of deviation	Set-point used as a reference for deviation of alarm evaluation.
		Low	Low temperature	–
		Very low	Very low temperature	–
		End limit	Process value outside of the practical range	–

Function	Icon	Meaning	Examples of use	Comment
Displaying trend pens		Present value	–	–
		Output value Very high limit, High limit, Very low limit, and Low limit	–	–
		Setpoint	–	–
		Deviation	–	–
Displaying labels	WI1101	Label of symbols	–	Only if labels are made visible.

Displaying Genies

Overview

The label that identifies each dynamic object inserted in a page can be shown or hidden during runtime.



NOTE: To hide or show the labels, click **Show/Hide Labels** option on the header of the runtime, when the **Label/Disable** is entered as `FALSE` in the property dialog box of the genie.

Faceplate

Overview

This section provides general information about the representation of the faceplates linked to genies.

You can access a faceplate by clicking the genie during operation.

Faceplates allow viewing and controlling the corresponding Control block.

Tab Icons

Overview

Faceplates consists of tabs that are grouped according to their functionalities provided by the associated Control block during operation.

Each tab is represented by an icon that you can click to display the information related to it.

Certain tabs are and become available if the control module features the corresponding element and the element is selected.

NOTE: The order of the tabs in a faceplate can be adapted to suit the application without impact on functionality. The tabs order shown in documentation images may differ from that displayed at runtime to the operator.

Description

The table describes the functions that are available in each tab:

Tab	Icon	Functions	Examples of use
Operation		• Current status (present value, setpoint, output value) • Owner change • Operating mode change • Resetting • Configuration of alarms at control level • Local panel monitoring	• Operator/program • Manual/automatic • PID controller setpoint (`SP`) change in auto mode and output change in manual mode.
Interlocks		• Interlock condition status • Bypassing and/or manual resetting of interlock conditions. • Navigate to the equipment of the detected error condition.	Interlock condition of the control module.
`Failure`		• Status of detected error conditions. • Bypassing of detected error conditions. • Navigate to the equipment of the detected error condition. • Navigate to the initial conditions of pumpset pattern and flow control.	Detected error condition of the control module.
Engineering		Changing settings or parameters.	• Adjustment of PID control parameters. • Activating the simulation mode.
Alarm		• Acknowledgment of alarms • Configuration of alarms at supervision level. • Navigate to the equipment of the alarm condition.	• Acknowledge alarm
Event		Logs the sequence of events	Change of owner
Direction1Inter-lock		• Direction1 interlock condition status • Bypassing and/or manual resetting of direction1 interlock conditions. • Navigate to the equipment of the detected error condition.	Interlock the forward direction of motor.

Tab	Icon	Functions	Examples of use
Direction2Inter-lock		• Direction2 interlock condition status • Bypassing and/or manual resetting of direction2 interlock conditions. • Navigate to the equipment of the detected error condition.	Interlock the reverse direction of motor.
Measures		Monitors the current values of parameter groups.	Monitoring the current value of voltage, current, frequency, etc.

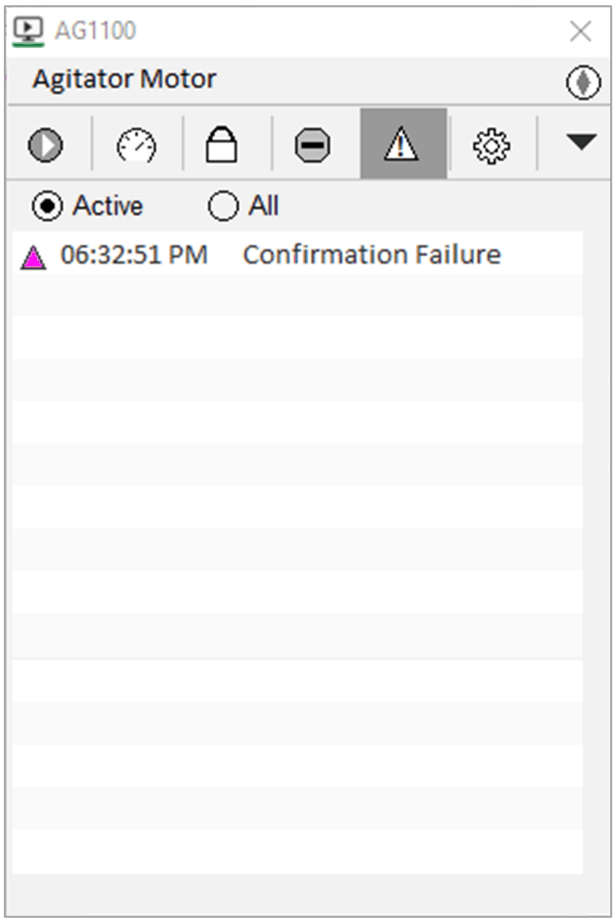
Alarm Tab

Overview

The alarm tab is available in each faceplate. It shows all the configured and active alarms that are associated with the control module and its children.

Alarm Tab Representation

The figure shows an example of an alarm tab:



Tab Icons

Shelved/Disabled Alarm

If any alarm of the control module is in an unacknowledged state, then action required condition is displayed on the alarm tab with flashing orange abnormal indication.

Unacknowledged Alarm

If any alarm of the control module is in an unacknowledged state, then action required condition is displayed on the alarm tab with flashing orange abnormal indication.

Alarm List

Display order

The alarms in the alarm tab are sorted based on the following order:

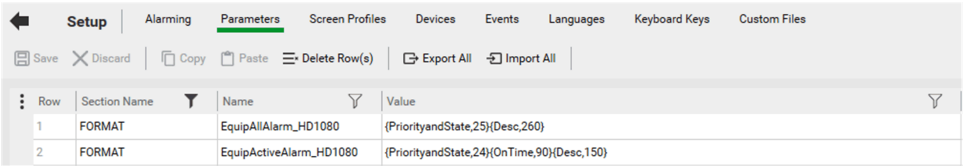
- Alarm Unacknowledged + Active
- Alarm Unacknowledged + Inactive
- Alarm Acknowledged + Active

NOTE:

- The order of displaying an alarm in each state is based on the priority (critical to low).
- The same priority and same state alarms are sorted based on the descending time order.

Display Format

User can manage the display format of an alarm list details by overwriting the **Value** field of `DefaultGPLEquipAllActiveAlarm_HD1080` and `DefaultGPLEquipActiveAlarm_HD1080` available in the **Parameter** tab (Setup > Parameters). Parameters can be overwritten by configuring the same parameter with required value in the Plant SCADA project. Refer to the figure.



For example, if user wants to display the category of individual alarm in the third column of an alarm list shown on the faceplate, overwrite the value of the parameter by `{PrioritiyandState,25}{Desc,260}{category,20}`.

Selection

User can select one of the following alarm list display:

- **Active**: Alarm associated to control module.
- **All**: Configured alarms of the control module.

  NOTE: When the *EnableChildSelection* parameter (Setup > Parameters) under section `GPL.Alarm` is configured as `TRUE`, user can select to display the alarms in the child equipment.

Action

Following are the possible actions:

- Acknowledge
- Shelve For
- Shelve Until
- Restore
- Navigate

  NOTE: Navigation (sets the context to the selected child) is available for the associated child equipment alarms.

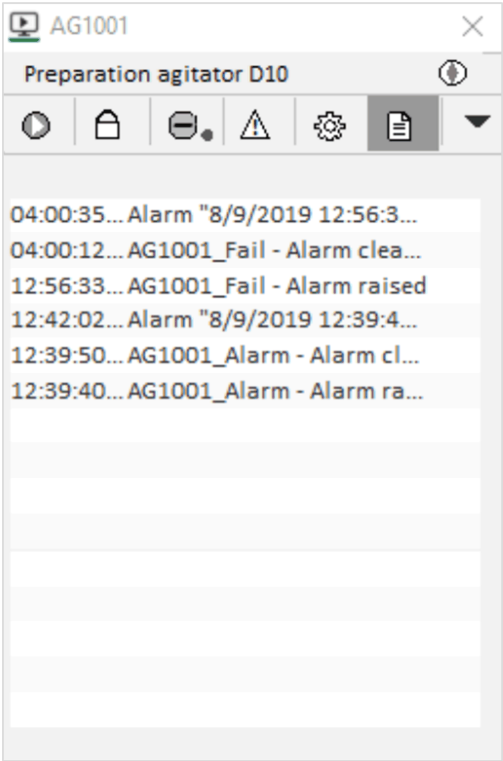
Event Tab

Overview

The Event tab is available in each faceplate. It logs the sequence of events associated to each alarm.

Event Tab Representation

The figure shows an example of an event tab:



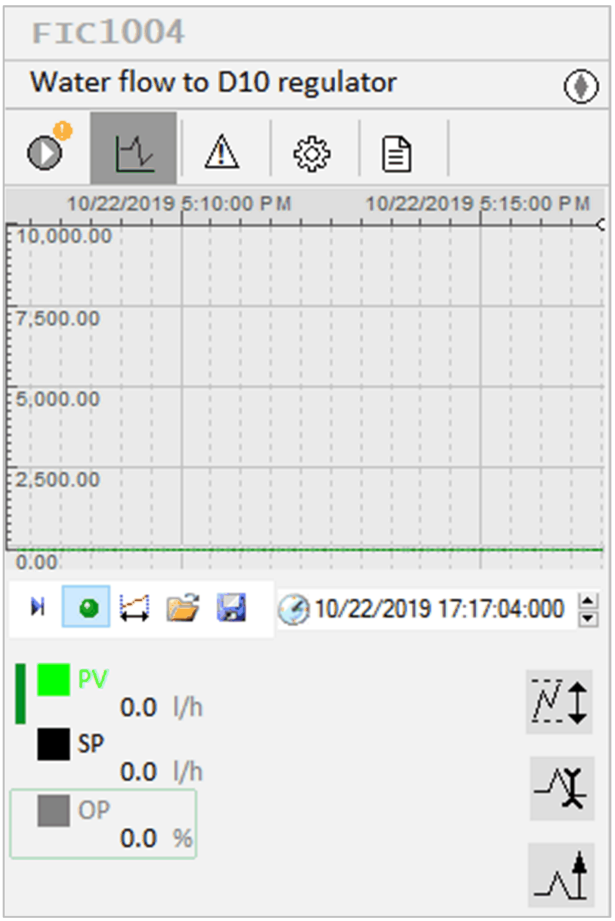
Trend Tab

Overview

The Trend tab is available in faceplate for signal processing families. It displays the logged data of parameters associated to the signal processing family.

Trend Tab Representation

The figure shows an example of a trend tab:



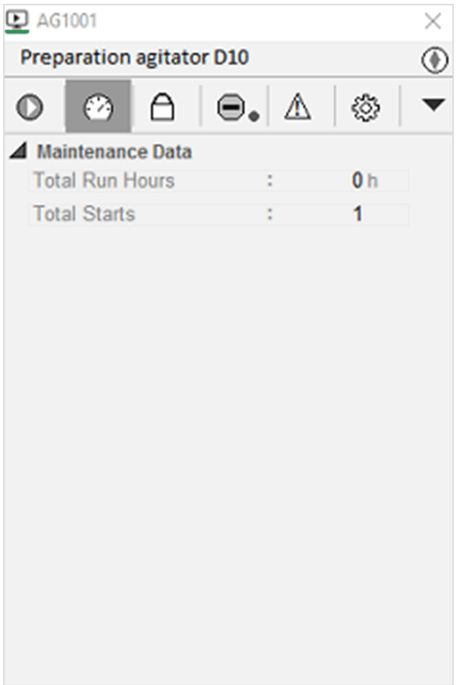
Measures Tab

Overview

The measures tab allows you to observe the status and monitor the current values of parameter groups (such as current, voltage, frequency, device status, diagnostics, etc) from the process templates.

Measures Tab Representation

The figure shows an example of a measures tab:



The table describes the control function of maintenance data group:

Label	Item name	Description
Maintenance Data		
Total Run Hours	MntResetHoursCount-er	Total run hours continuous counter.
Total Starts	MotMntResetManeu-versCounter	Total starts.

For more details about parameter groups, refer to the **Measures Tab** section in respective supervision chapter.

Color Codes

Dynamic Variable Status

Description

The table describes the possible statuses of dynamic variables and the colors that are used to represent them in elements of Supervision components:

Function	Color	Status	Example of use
Discrete module	Grey	Active	Active running order Active running confirmation Active open valve order Open valve
	White	Passive	Idle running order Idle running confirmation Active close valve order

Function	Color	Status	Example of use
			Closed valve
	Black	Inconsistency	Open-valve and closed-valve limit switches are active simultaneously
Numeric indicators	400.0	Present value	Current temperature value
	400.0	Setpoint value	Target temperature value
	100.0	Output value	Control valve position

Optional Services

What's in This Part

Failures 74
Interlocks 81
Maintenance 89
Analog Alarm 108

Overview

This part describes the optional services that are available from templates.

Failures

What's in This Chapter

Template 74
Control 75
Supervision.....78

Template

Description

General Description

The `$CONDSUMGP` template provides the service to detect errors associated with equipment. A maximum of 15 conditions can be detected. For example, interlocks or initial conditions. The template allows evaluating the conditions by using the `OR` logic. Each of them can require individual resetting or be ignored according to the template configuration.

You can use this template as a supplement to other General Purpose Library templates to evaluate their interlocking inputs.

Function Description

The following table describes the main functions of the template:

Function	Description
Condition Evaluation	The template performs the logical or evaluation of the given inputs and outputs the result.
Manual Reset	Conditions configured this way require resetting from the monitoring system.
Bypass	Enables ignoring conditions (one by one) when performing the logical evaluation.

Parameters

Parameters

The table describes the control and supervision parameters:

Section	Parameter description	Type	Default value	Additional remarks
Control parameter				
Description	**Condition {xx}**	`String`	–	Description that appears on the **Failures** tab of the faceplate when corresponding detected error condition arises.
	Group description		*(Failures)*	Group description that appears on the **Failures** tab of the faceplate when corresponding detected error condition arises.
Reset Required	**Condition {xx}**	`Boolean`	*FALSE*	When set to true, requires a manual reset.
Disable Bypass	**Condition {xx}**			Disables bypass condition on **Failures** tab.
NOTE: {xx} corresponds to the number of the detected condition from 1 to 15.				

Section	Parameter description	Type	Default value	Additional remarks
Supervision parameter				
Historize	**Conditions**	`Boolean`	*FALSE*	When set to true, historizes detected error condition activity.

Available alternate services:

Service	Description
Condsum2On	Detected error On conditions with 2 groups (15 conditions each).
Condsum1OnAlarm	Detected error On alarm conditions with 1 group (15 conditions).
Condsum2OnAlarm	Detected error On alarm conditions with 2 groups (15 conditions each).

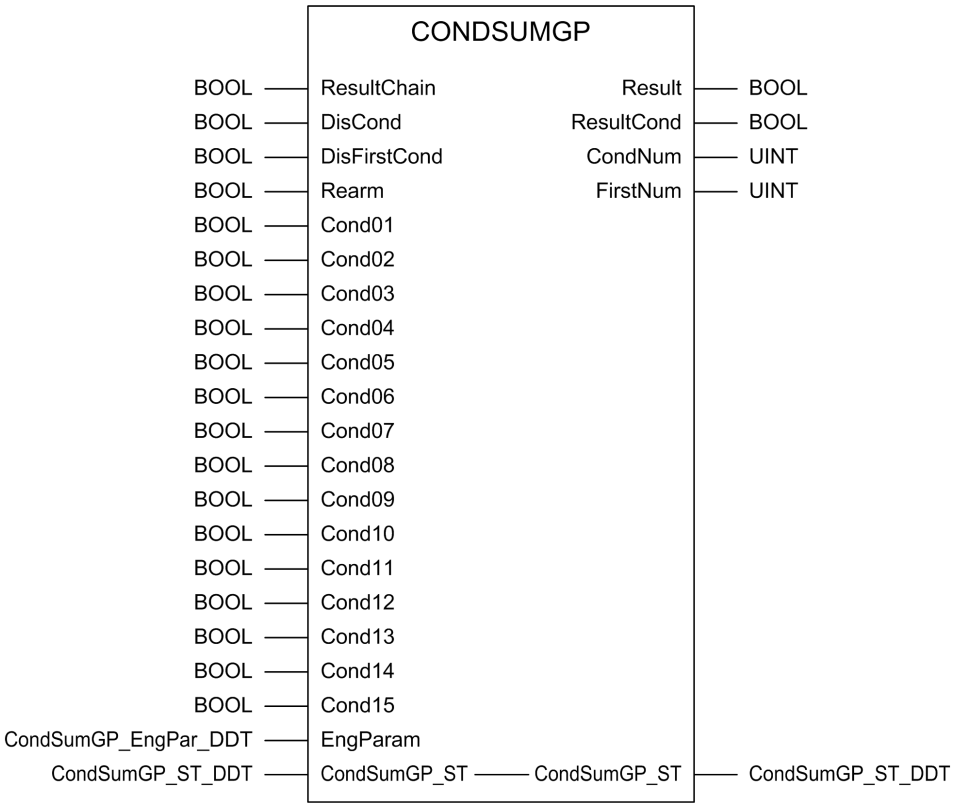
These services are available as control and supervision composite (`UC` and `CD` respectively).

Control

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



NOTE: The above representation is also applicable for `CONDSUMOFFGP`.

Inputs

Input Parameter Description

Name	Type	Description
ResultChain	BOOL	Chain condition status (1 = Condition active).
DisCond	BOOL	Disable all conditions (1 = Disable).
DisFirstCond	BOOL	Disable first up condition detected (1 = Disable).
Rearm	BOOL	Enables the device to reset (1). The DFB sets the signal to 0 after each execution and sets the signal indicating that resetting is required (CFGW.Rearm) to 0.
Cond{x}	BOOL	1 = Condition is activated.
NOTE: {x} represents the condition number from 01 to 15.		

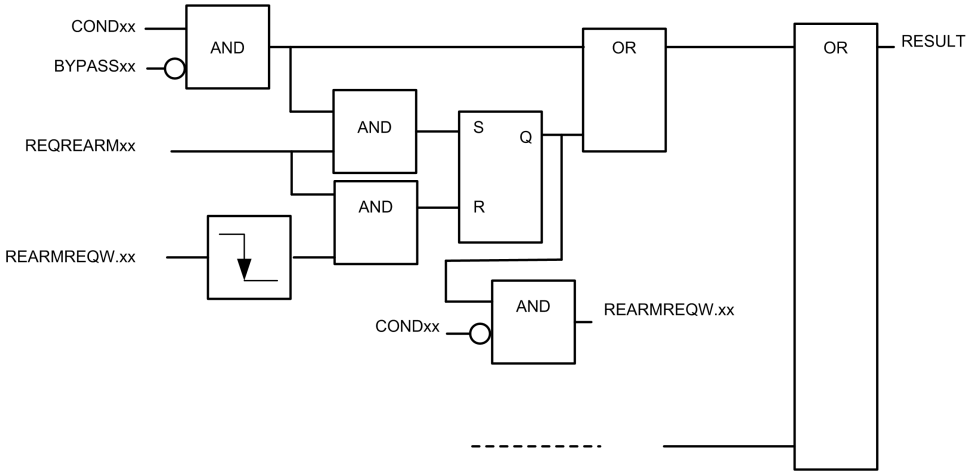
CondSumGP_EngPar_DDT Type

Name	Bit Name	Bit	Type	Description
RearmReqCW	–	–	BYTE	1 = Enables configuring that a manual reset is required when the corresponding input condition (Cond{x}) is activated.
	RearmReqC{x}	{y}	BOOL	1 = Condition 01 requires rearm once condition returns to normal.
BypassDisCW	–	–	BYTE	1 = Bypassing disabled. Makes it possible to disable bypassing of the corresponding interlock conditions (Cond{x}).
	BypassDis{x}	{y}	BOOL	Disable bypass for condition 01 (1 = Bypass disabled).
NOTE: {x} represents the condition number from 01 to 15 and {y} represents the bit number from 0 to 14.				

Outputs

Output Parameter Description

Name	Type	Description
Result	BOOL	1 = Summarizes the result of the logical OR evaluation of the conditions according to the following logic diagram.
Result-Cond	BOOL	Result of all the conditions in this DFB, excluding chain condition (1 = At least one condition in this block is active or waiting for rearm).
CondNum	UINT	Number of the active condition, which considers an ascending order (0 when no conditions are active).
First-Num	UINT	Number of the active first condition detected (0 when no active first condition detected).



Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
CondSumGP_ST	CondSumGP_ST_DDT	Data structure that is used as an interface with the monitoring subsystem.

CondSumGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
CondW	–	–	WORD	Read-only access Input conditions (refer to the Cond{x} inputs, page 76) and their evaluation summary (refer to the Result output pin, page 76.
	Cond{x}	{y}	BOOL	1 = Condition 01 is active.
	Result	15	BOOL	Refer to the Result output pin, page 76.
FirstCondW	–	–	WORD	First condition
	FirstCond{x}	{y}	BOOL	1 = Condition 01 is the first condition.
BypassW	–	–	WORD	Read/write access Indicates the bypass signals set from the monitoring system.
	Bypass{x}	{y}	BOOL	1 = Condition 01 bypassed.
	DisCond	15	BOOL	1 = All conditions disabled.
RearmReqW	–	–	WORD	Read/write access Indicates the signals requiring reset (1) set from the DFB and forced resetting (0) from the monitoring system.
	RearmReq{x}	{y}	BOOL	1 = Condition 01 require rearm.
NOTE: {x} represents the condition number from 01 to 15 and {y} represents the bit number from 0 to 14.				

The following table describes the CondSumGP_ST.CondW word:

Bit	Description
0	Cond01
1	Cond02
...	...

Bit	Description
14	Cond15
15	Result

The following table describes the `CondSumGP_ST.BypassW` word:

Bit	Description
0	Bypass01
1	Bypass02
...	...
14	Bypass15

The following table describes the `CondSumGP_ST.RearmReqW` word:

Bit	Description
0	RearmReq01
1	RearmReq02
...	...
14	RearmReq15

Public Variables

Public Variable Description

Variable	Type	Description
`SC`	`CondSumGP_SC_DDT`	Provides the frequently needed data to monitor the DFB status from the sequential control.

`CondSumGP_SC_DDT` Type

Name	Type	Description
`Result`	`BOOL`	Read-only access Refer to the `Result` output pin, page 76.

Supervision

Failures Tab

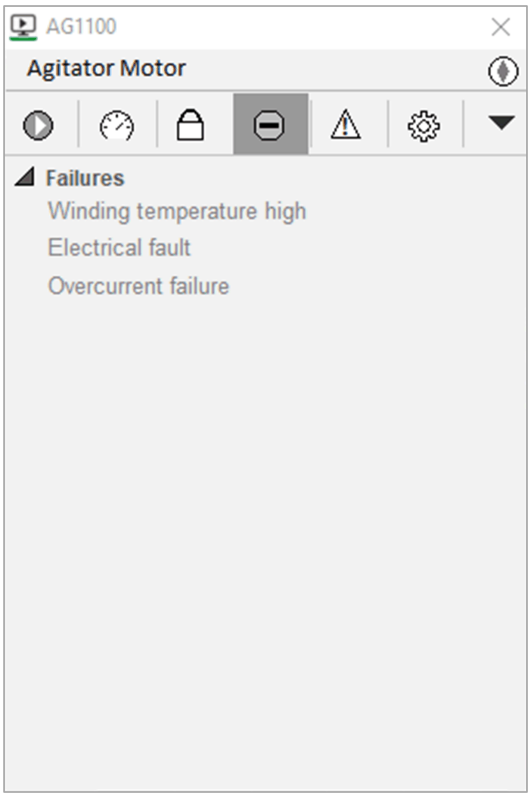
Overview

The optional **Failures** tab is available on certain faceplates and provides the following functions depending on the configuration of the instance in EcoStruxure Process Expert:

- Viewing and resetting of individual detected error conditions.
- Bypassing of a detected error condition.
- Navigation, if configured in the control block.

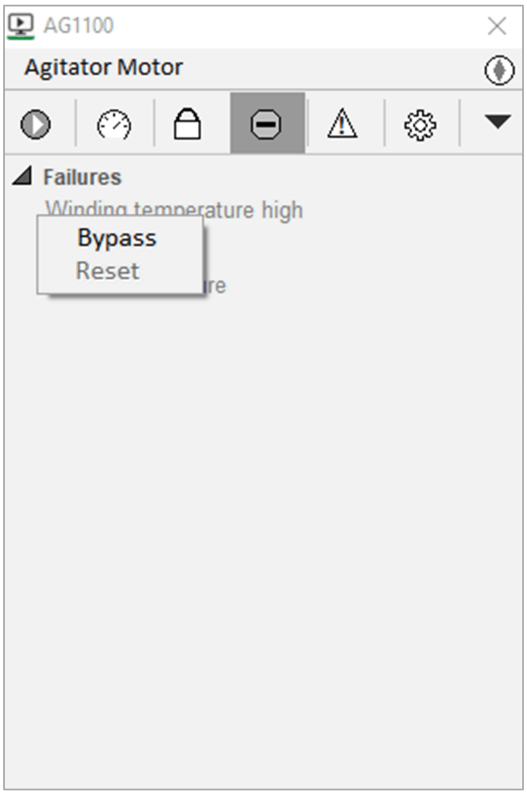
Failures Tab Representation

The figure shows an example of the **Failures** tab with group:



There are *4* different detected error groups available and each group contains the *15* detected error conditions. Based on the configuration from Process Expert at the instance level, groups and conditions are visible in **Failures** tab.

The figure shows an example of a group context menu commands for an individual condition:



Bypassing Detected Error Conditions

Individual detected error conditions can be bypassed by selecting **Bypass**. The default privilege for selecting **Bypass** is *operator*.

Manual Resetting of Detected Error Conditions

The reset is available only if the detected error condition has been cleared and depending on the configuration made in the controller. The default privilege for selecting **Reset** is *operator*.

When the reset is selected and this is the only detected error condition, then the current setpoint shown in the operator tab is initiated.

Navigating to the Equipment Signaling a Detected Error Conditions

Navigation is available if the equipment is configured for detected error condition. When the navigate command is selected, the configured equipment faceplate will open.

Interlocks

What's in This Chapter

Template 81
Control 82
Supervision.....87

Overview

This chapter describes the control and supervision functionality of the template that provides the interlock service.

Template

Description

Interlocks are process conditions which require the equipment to start or stop. When an interlock condition becomes active it enables the interlock block output which forces the control module output to the desired state. The interlock conditions identify the first condition to set the interlock to true and displays the cause of the interlock to the operator. Interlock conditions can be disabled based on process conditions.

Other interlock facet templates supporting higher interlock counts and different interlock types are available within the library.

Parameters

The table describes the control and supervision parameters:

Section	Parameter description	Type	Default value	Additional remarks
Control parameter				
Description	**Condition{xx}**	`String`	–	Description that appears on the interlock tab of the faceplate. To configure the description, refer to the **Configuring Description**.
	Group description		*(Interlocks)*	Group description that appears on the interlock tab of the faceplate when corresponding detected interlock condition arises.
Reset Required	**Condition{xx}**	`Boolean`	*FALSE*	When set to true, requires rearming the interlock condition detection.
Disable Bypass	**Condition{xx}**			When set to true, enables bypassing an interlock condition.
NOTE: {xx} corresponds to the number of the interlock condition from 1 to 7.				
Supervision parameter				
Historize	**Conditions**	`Boolean`	*FALSE*	When set to true, historizes interlock condition activity.

Configuring the Description

The description format is, <Asset Instance Id which generates this `condition>#<condition description>`

Asset instance id is required only if user wants to navigate to the faceplate of the asset which generated the interlock condition. **#** is separator between the instance id and the description and needs to be added only if instance id is configured.

The table lists examples of valid descriptions:

Description format	Linked asset instance	Condition description which will appear on faceplate
V1101 # (valve not open)	V1101	valve not open
V1101 # (valve not open)	V1101	valve not open
V1101 # valve not open	–	valve not open
valve not open	–	valve not open
(valve not open)	–	valve not open
NOTE: () syntax is for support of multi-language in Citect.		

Available alternate services:

Service	Description
Interlock1Off	Interlock Off conditions with 1 groups (7 conditions each).
Interlock1On	Interlock On conditions with 1 groups (7 conditions each).
Interlock2Off	Interlock Off conditions with 2 groups (7 conditions each).
Interlock2On	Interlock On conditions with 2 groups (7 conditions each).
Interlock4Off	Interlock Off conditions with 4 groups (7 conditions each).
Interlock4On	Interlock On conditions with 4 groups (7 conditions each).
Interlock8Off	Interlock Off conditions with 8 groups (7 conditions each).
Interlock8On	Interlock On conditions with 8 groups (7 conditions each).

These services are available as control and supervision composites (`UC` and `CD`).

Control

Description

General

The features of the `ILCKONGP and ILCKOFFGP` DFBs are similar to those of the `CONDSUMGP` DFB. The `ILCKONGP and ILCKOFFGP` DFBs are used to evaluate up to 7 conditions to implement interlocking strategies. Each DFB allows the logic `OR` evaluation of these conditions. Each of them can require individual resetting or be ignored (bypassed) according to the DFB configuration.

You can link this DFB to other DFBs of the General Purpose library to evaluate their interlocking inputs and positions.

The table compares the functions of the `CONDSUMGP` and `ILCKONGP/ILCKOFFGP` DFBs:

Function	`CONDSUMGP`	`ILCKONGP/ILCKOFFGP`
Maximum number of conditions	15	7
Condition prioritization	Y	Y
Evaluation of interlock positions	N	Y
Possibility of inhibiting bypassing on a condition-by-condition basis	Y	Y

Function Description

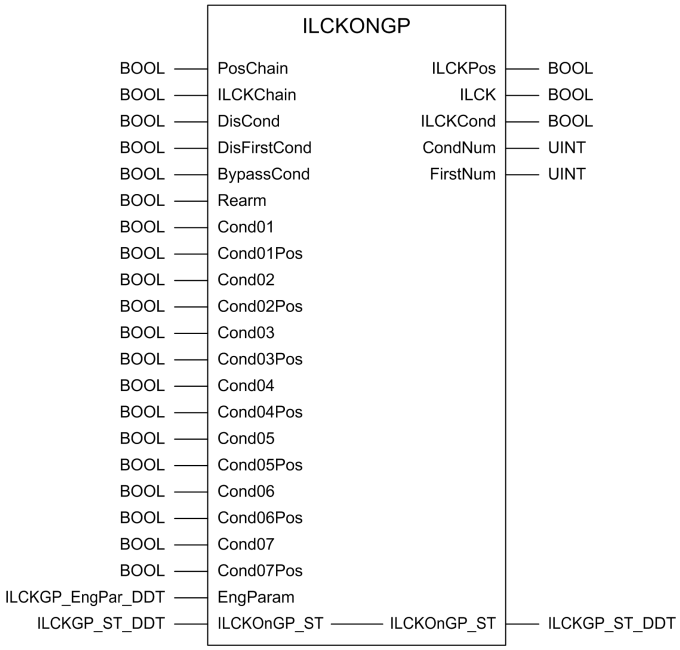
The main functions of the `ILCKONGP` and `ILCKOFFGP` DFBs are described in the following table:

Function	Description
Condition prioritization	The DFB performs the logical `OR` evaluation, giving Condition 1 priority over Condition 2, Condition 2 priority over Condition 3, and so on.
Evaluation of interlock positions	Each condition has a discrete position associated with it. The DFB evaluates the position of each condition based on the active conditions evaluated according to priority.
Interlocks Featuring Manual Resetting	Process conditions configured this way require resetting from the Supervision system.
Bypass	Allows configuring bypass interlock conditions one by one.
Active Interlock Indication	The DFB features output signals that report the state of each interlock condition after internal logic has been applied (resetting and bypassing). These signals can be used to implement program logic, or to select the analog position for other DFBs that require it (for example, `AOUTPUTGP`) by using an external multiplexer (for example, `ASELECT1`).

DFB Representation

Representation

The `ILCKONGP` DFB has been specifically designed for use with the FBD language of the controller.



NOTE: The above representation is also applicable for `ILCKOFFGP`.

Inputs

Input Parameter Description

Name	Type	Description
`PosChain`	`BOOL`	Position to be considered during chain condition.
`ILCKChain`	`BOOL`	Chain condition status (1= Condition active).

Name	Type	Description
DisCond	BOOL	Disable all conditions (1 = Disable).
DisFirstCond	BOOL	Disable first up condition detected (1 = Disable).
BypassCond	BOOL	Bypass all conditions (1 = Bypass).
Rearm	BOOL	Enables the device to reset (1). The DFB sets the signal to 0 after each execution and sets the signal indicating that resetting is required (CFGW.Rearm) to 0.
Cond{x}	BOOL	0 = Interlock condition activated
Cond{x}Pos	BOOL	Position to be considered during condition {x}.
EngParam	ILCKGP_ EngPar_DDT	Position to be considered during condition {x}.
NOTE: {x} represents from 01 to 07.		

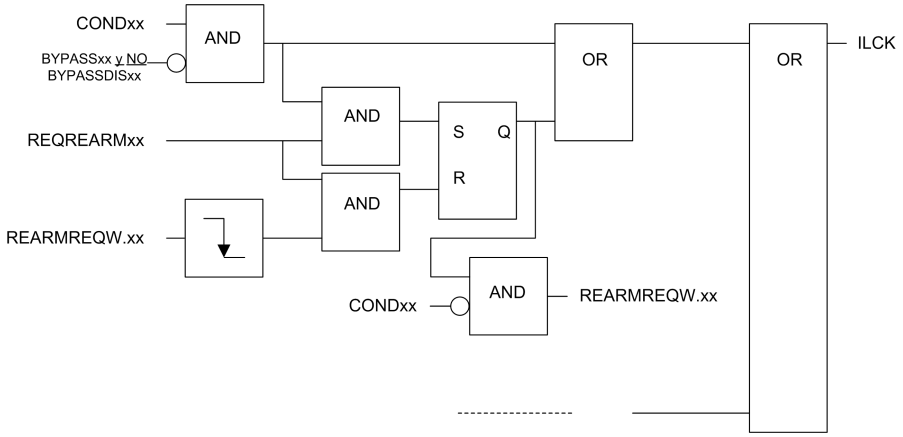
ILCKGP_EngPar_DDT Type

Name	Bit Name	Bit	Type	Description
RearmReq	–	–	BYTE	Require rearm.
	RearmReqC{x}	{y}	BOOL	1 = Condition 01 requires rearm once condition returns to normal.
BypassDis	–	–	BYTE	1 = Bypassing Disabled. Makes it possible to disable bypassing for the corresponding interlock conditions (Cond{x}).
	BypassDis{x}	{y}	BOOL	Disable bypass for condition 01 (1 = Bypass disable).
NOTE: {x} represents from 01 to 07 and {y} represents from 0 to 6.				

Outputs

Output Parameter Description

Name	Type	Description
ILCKPos	BOOL	1 = Indicates the interlock position (refer to the Cond{x}Pos, page 83 inputs) that is associated to the active interlock after priorities have been applied (refer to the ILCK output). If no interlocks are active, the ILCKPos output is set to 0 by default.
ILCK	BOOL	1 = Summarizes the status of the interlock after performing the logic OR evaluation of the conditions according to priority and the following logic: Conditions are applied with priority. Every condition is evaluated every time, but the position (refer to the ILCKPos output) is determined solely based on the first active interlock condition (the one with the highest priority is condition 01 while the one with the lowest priority is condition 07).
ILCKCond	BOOL	1 = Activated depending on the corresponding interlock condition that is active (considering both bypassing and whether or not the condition requires manual resetting). Each output is independent from the previous one so that the number of ILCKCond outputs activated corresponds to the number of simultaneous active interlocks.
ILCKNum	UINT	Indicates the index of the active interlock (1..7). Set to 0 by default if no interlocks are active.
FirstNum	UINT	Number of the active first condition detected (0 when no active first condition detected).



Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
ILCKOnGP_ST/ ILCKOffGP_ST	ILCKGP_ST_DDT	Data structure that is used as an interface with the monitoring subsystem.

ILCKGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
Cond	–	–	BYTE	Read-only access Conditions
	Cond{x}	Bit{y}	BOOL	1 = Condition 01 is active.
	Result	7	BOOL	Result of all the conditions in this DFB, excluding chain condition (1 = At least one condition in this block is active or waiting for rearm).
FirstCond	–	–	BYTE	Read-only access First condition
	FirstCond{x}	Bit{y}	BOOL	1 = Condition 01 is the first condition.
Bypass	–	–	BYTE	Read/write access Bypass conditions (Read/Write).
	Bypass{x}	Bit{y}	BOOL	1 = Condition 01 bypassed.
	DisCond	7	BOOL	1 = All conditions disabled.
RearmReq	–	–	BYTE	Read/write access Requires rearm.
	RearmReq{x}	{y}	BOOL	1 = Condition 01 requires rearm.
NOTE: {x} represents the number of the interlock condition from 01 to 07 and {y} represents bit numbers from 0 to 6.				

ILCKGP_ST.Cond Word Structure

Read-only access.Input conditions (refer to the Cond{x}, page 83 inputs) and their evaluation summary (refer to the ILCK, page 84 output). The following table describes the ILCKGP_ST.Cond word:

Bit	Description
0	Interlock 01. Refer to the ILCK01, page 84 output pin.
1	Interlock 02. Refer to the ILCK02, page 84 output pin.
...	...
6	Interlock 07. Refer to the ILCK07, page 84 output pin.
7	Resulting interlock. Refer to the ILCK, page 84 output pin.

ILCKGP_ST.Bypass Byte Structure

Read/write access. Bypass signals set from the monitoring system. The following table describes the ILCKGP_ST.Bypass byte:

Bit	Description
0	Active bypass (1) for interlock 01
1	Active bypass (1) for interlock 02
...	...
6	Active bypass (1) for interlock 07.

ILCKGP_ST.RearmReq Byte Structure

Read/write access. Signals requiring resetting (1) set from the DFB and forced resetting (0) signals set from the monitoring system. The following table describes the ILCKGP_ST.RearmReq byte:

Bit	Description
0	Interlock 01 requires resetting (1) or does not require resetting (0). Interlock 01 is reset by setting it to 0.
1	Interlock 02 requires resetting (1) or does not require resetting (0). Interlock 02 is reset by setting it to 0.
...	...
6	Interlock 07 requires resetting (1) or does not require resetting (0). Interlock 07 is reset by setting it to 0.

Public Variables

Public Variable Description

Variable	Type	Description
SC	ILCKGP_SC_DDT	Provides the frequently needed data to monitor the status of the DFB from the sequential control.

ILCKGP_SC_DDT Type

Name	Type	Description
ILCK	BOOL	Read-only access Refer to the ILCK output pin, page 84.
ILCKPos	BOOL	Read-only access Refer to the ILCKPos output pin, page 84.

Supervision

Interlocks Tab

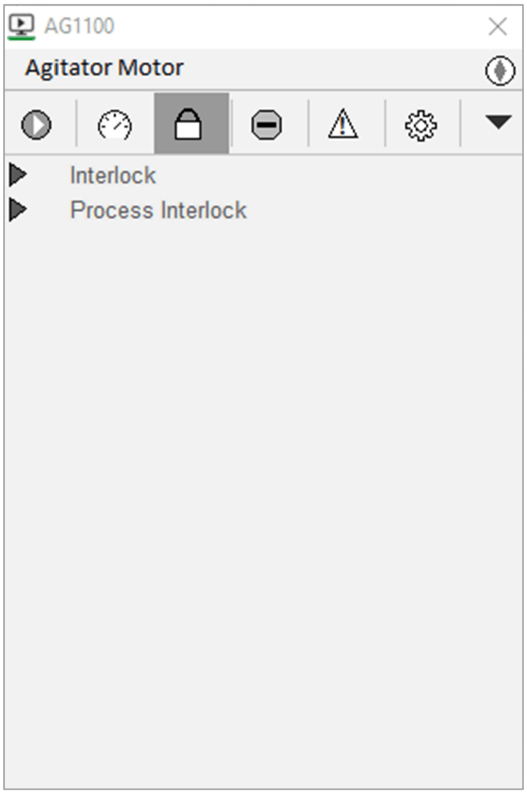
Overview

The optional **Interlocks** tab is available on certain faceplates and provides the following functions depending on the configuration of the instance in EcoStruxure Process Expert:

- Viewing and resetting of individual interlock conditions.
- Bypassing of interlock condition.
- Navigation, if configured in EcoStruxure Process Expert.

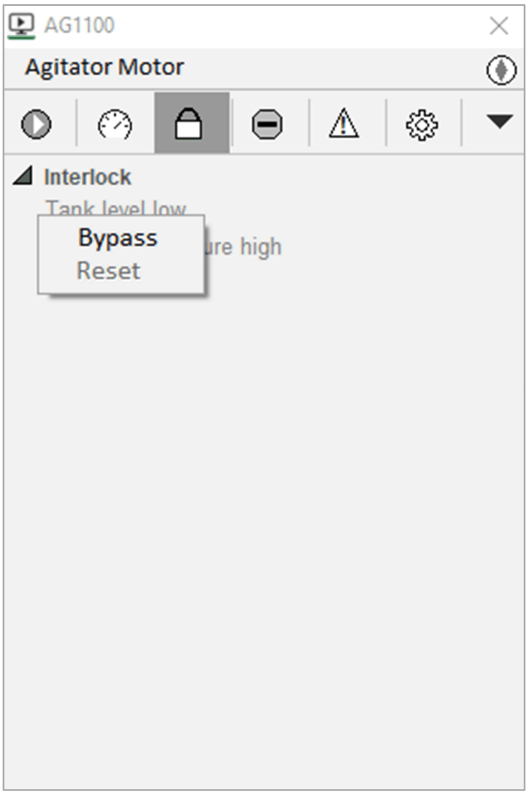
Interlocks Tab Representation

The figure shows an example of the **Interlocks** tab with groups:



There are *4* different interlock groups available and each group contains the *7* interlock conditions. Based on the configuration from Process Expert at instance level, groups and conditions are visible in **Interlocks** tab.

The figure shows an example of a group with context menu commands for an individual interlock condition:



Bypassing an Interlock Condition

Individual interlock conditions can be bypassed by selecting **Bypass**. The default privilege for selecting **Bypass** is *operator*.

Manual Resetting of an Interlock Condition

The reset is available only if the interlock condition has been cleared depending on the configuration made in the controller. The default privilege for selecting **Reset** is *operator*.

When the reset is selected and this is the only interlock condition, then the current setpoint shown in the operator tab is initiated.

Navigation to the Equipment that Represents an Interlock Condition

Navigation is enabled, if the equipment is configured for interlock condition. When the **Navigate** command is selected, the configured equipment faceplate appears.

Maintenance

What's in This Chapter

Template 89
Control 90
Supervision 92
Template - Alternate Services (Advance Motor Maintenance) 93
Control - Alternate Services (Advance Motor Maintenance) 94
Supervision - Alternate Services (Advance Motor Maintenance) 97
Template - Alternate Services (Advance Valve Maintenance) 101
Control - Alternate Services (Advance Valve Maintenance) 101
Supervision - Alternate Services (Advance Valve Maintenance) 104

Template

Description

The maintenance facet displays run-hours and counts operations. The maintenance values are available on the **Measures** tab of the faceplate. The maintenance counters can be reset by clicking on the value displayed.

If the counters are not reset before the maximum value is reached a rollover alarm will be generated.

Note:

- In **Engineering** Tab, motor start count alarm (`Warning Limit`, `Alarm Limit`) should be configured greater than zero to get motor start count alarm and motor start count `warning`.
- In **Engineering** Tab, valve operation count alarm (`Warning Limit`, `Alarm Limit`) should be configured greater than zero to get valve operation count alarm and valve operation count `warning`.

Additional maintenance facet templates providing more maintenance functionality are available within the library.

Parameters

Control

The table describes the Control parameters:

Section	Parameter description	Type	Default value	Additional remarks
Configuration	**Device maintenance signal**	`Enum`	*Feedback (0)*	Device maintenance signal

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Alarm	**Counter overflow severity**	`Enum`	*Low (4000)*	The default alarm severity is medium in all the templates.
	Counter overflow group	`String`	*Maintenance*	The default severity of all the maintenance alarm is low in all the templates.
Historize	**Counters**	`Boolean`	*FALSE*	When set to true, historize counters.

Control

Description

General

The DEVMNTGP DFB is used to provide data that is useful for device maintenance.

You can use this DFB as a supplement to the MOTORGP DFB or independently. This DFB features totalizing functions for hours of operation and switching operations.

Function Description

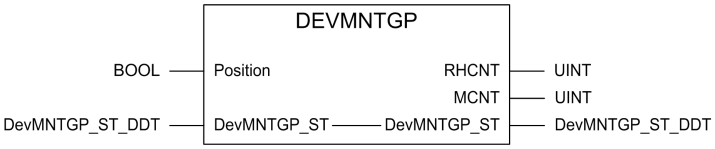
The main functions of the DFB are described in the following table:

Function	Description
Switching Operation Totalization	Counts the number of switching operations that have been performed on the device since the totalizer was last reset.
Hours of Operation Totalization	Counts the number of hours that the device has been in operation since the totalizer was last reset.
Reset	The DFB allows to resetting the described totalizers individually.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
Position	BOOL	1 = ON (running, open, and so on). Provides the device status information with which the outputs are calculated.

Outputs

Output Parameter Description

Parameter	Type	Description
RHCNT	UINT	Totalizes the number of the device hours of operation. The calculation is done based on the position indicator of the device shown by the Position input.
MCNT	UINT	Total number of switching operations of the device. The calculation is done based on the position indicator of the device shown by the Position input. The count is incremented each time a rising edge is detected.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
DevMNTGP_ST	DevMNTGP_ST_DDT	Data structure that is used as an interface with the monitoring subsystem.

DevMNTGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the DFB status usually used from the monitoring subsystem. Read-only access to the data contained in this word.
	HOV	Bit0	BOOL	HOV is activated when the hours counter exceeds the maximum representable value (65535). It is deactivated by using DevMNTGP_ST.STW.RSTH.
	MOV	Bit1	BOOL	MOV is activated when the switching operations counter exceeds the maximum representable value (65535). It is deactivated by using DevMNTGP_ST.STW.RSTM.
CFGW	–	–	WORD	Provides the means to control the DFB. Read-only access to the data contained in this word.
	RSTH	Bit0	BOOL	Enables resetting the hours of operation totalizer. Also resets the internal seconds subtotal from the last hour of operation fraction and from the DevMNTGP.STW.HOV signal. The DFB sets this signal to 0 after it has been processed.
	RSTM	Bit1	BOOL	Enables resetting the switching operation totalizer. Also resets the DevMNTGP.STW.MOV signal. The DFB sets this signal to 0 after it has been processed.
RHCNT	–	–	UINT	Read-only access Refer to the RHCNToutput pin, page 91.
MCNT	–	–	UINT	Read-only access Refer to the MCNT output pin, page 91.

Public Variables

Public Variable Description

Variable	Type	Description
SC	DevMNTGP_SC_DDT	Provides the frequently needed data to monitor the DFB status from the sequential control.

DevMNTGP_SC_DDT Type

Name	Type	Description
RHCNT	UINT	Read-only access Refer to the DevMNTGP_ST.RHCNT output pin, page 91.
MCNT	UINT	Read-only access Refer to the DevMNTGP_ST.MCNT output pin, page 91.

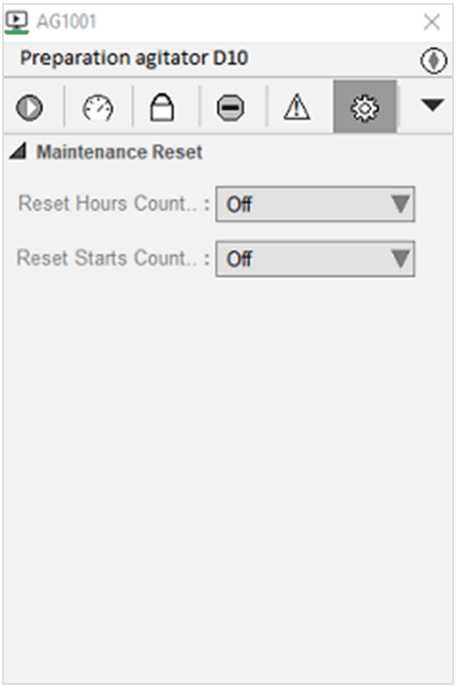
Supervision

Maintenance

Overview

The maintenance tab shows all the control module when it is in maintenance state.

The figure shows the maintenance group in **Engineering** tab:

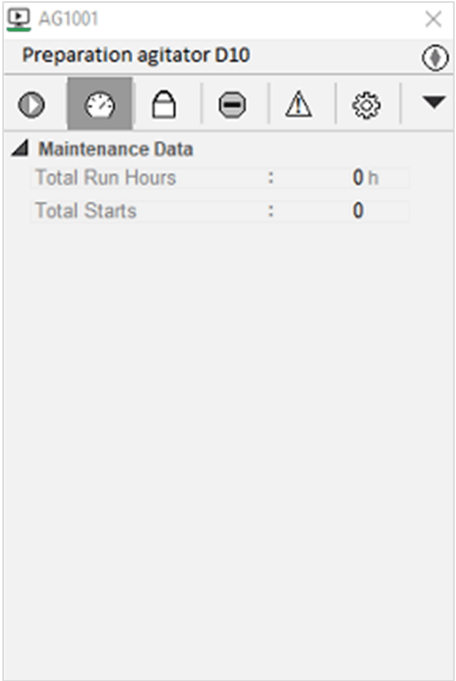


The table describes the control function of maintenance reset group:

Label	Item name	Description
Maintenance Reset		
Reset Hours Count	{x} MntResetHoursCount-er	Reset the hours counter.
Reset Starts Count	{x} MotMntResetManeu-versCounter	Reset the operations counter.

NOTE: {x} represents the Mot, Vlv, or Dev based on equipment type.

The figure shows the maintenance group in **Meausures** tab:



The table describes the control function of maintenance data group:

Label	Item name	Description
Maintenance Data		
Total Run Hours	{x} MotMntTotalRunHours	Total hours of operation.
Total Starts	{x}MotMntTotalStarts	Total starts.

NOTE: {x} represents Mot, Vlv, or Dev based on equipment type.

Template - Alternate Services (Advance Motor Maintenance)

Maintenance

Description

The maintenance facet displays run-hours and counts operations. The maintenance values are available on the **Measures** tab of the faceplate. The maintenance counters can be reset by clicking on the value displayed.

If the counters are not reset before the maximum value is reached a rollover alarm will be generated.

NOTE: In Engineering Tab, motor start count alarm (`Warning Limit`, `Alarm Limit`) should be configured greater than zero to get start count alarm and start count warning.

Additional maintenance facet templates providing more maintenance functionality are available within the library.

Parameters

Supervision

The table describes the supervision parameters:

Section	Parameter	Type	Default	Description
Alarm	`Counter overflow of alarm severity`	*Enum*	Low (4000)	The default alarm severity is medium in all the templates.
	`Counter overflow of alarm group`	*String*	Maintenance	The default severity of all the maintenance alarm is low in all the templates.
Historize	`Enable historize counters`	*Boolean*	False	When set to true historizes counters.
	`Enable historize status/commands`			Enables historize status or commands.

Control - Alternate Services (Advance Motor Maintenance)

Description

General

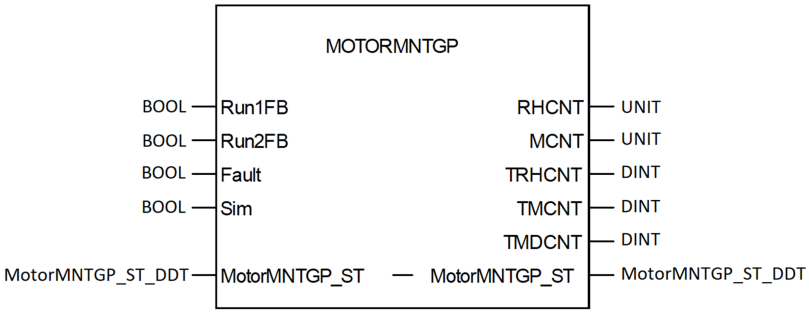
The `MOTORMNTGP` DFB is used to provide data that is useful for device maintenance.

You can use this DFB as a supplement to the `MOTORGP` DFB or independently. This DFB features a totalizing function for hours of operation and switching operations.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
Run1FB	BOOL	Actual primary position of device.
Run2FB	BOOL	Actual secondry position of device. **NOTE:** Run2FB link is removed from motorGP template. This is applicable only for motorGP.
Fault	BOOL	Device detected fault
Sim	BOOL	Enables to simulation mode.

Outputs

Output Parameter Description

Parameter	Type	Description
RHCNT	UINT	Totalizes the number of the device hours of operation. The calculation is done based on the position indicator of the device shown by the Position input.
MCNT	UINT	Totalizes the number of switching operations that the device has performed. The calculation is done based on the position indicator of the device shown by the Position input. A new switching operation is added each time a rising edge is detected.
TRHCNT	DINT	Total run hours continuous counter.
TMCNT	DINT	Total operations continuous counter.
TMDCNT	DINT	Maintenance days continuous counter.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
MotorMNTGP_ST	MotorMNTGP_ST_DDT	Data structure that is used as an interface with the monitoring subsystem.

MotorMNTGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the DFB status usually used from the monitoring subsystem. Read-only access to the data contained in this word.
	RHOV	Bit0	BOOL	1 = Running hours counter overflow bit.
	MOV	Bit1	BOOL	1 = Manuevers counter overflow bit.
	SHOV	Bit2	BOOL	1 = Stopping hours counter overflow bit.
	PRHOV	Bit3	BOOL	1 = Primary running hours counter overflow bit.
	SRHOV	Bit4	BOOL	1 = Secondary running hours counter overflow bit.
	PMOV	Bit5	BOOL	1 = Primary maneuvers counter overflow bit.

Name	Bit Name	Bit	Type	Description
	SMOV	Bit6	BOOL	1 = Secondary maneuvers counter overflow bit.
	FSOV	Bit7	BOOL	1 = Stopped with detected fault counter overflow bit.
	RDOV	Bit8	BOOL	1 = Reset days from last reset counter overflow bit.
	TRHOV	Bit9	BOOL	1 Total run hours continuous counter overflow bit (continous counters).
	TMOV	Bit10	BOOL	1 = Total manuvers counter overflow bit (continous counters).
CFGW	–	–	WORD	Provides the means to control the DFB. Read-only access to the data contained in this word.
	RSTCNT	Bit0	BOOL	1 = Reset bit for counters.
	RSTTCNT	Bit1	BOOL	1 = Reset bit for total counters (continous counters).
RHCNT	–	–	UINT	Refer to the RHCNT output pin, page 91.
SHCNT	–	–	UINT	Stopped hours counter.
PRHCNT	–	–	UINT	Primary direction running hours.
SRHCNT	–	–	UINT	Secondary direction running hours.
MCNT	–	–	UINT	Refer to the MCNT output pin, page 91.
PMCNT	–	–	UINT	Primary direction maneuvers numbers.
SMCNT	–	–	UINT	Secondary direction maneuvers numbers.
FSCNT	–	–	UINT	The number of times the device has stopped with detected fault.
ResetDays	–	–	UINT	The number of days since last reset.
TRHCNT	–	–	DINT	Refer to the TRHCNT output pin, page 91.
TMCNT	–	–	DINT	Refer to the TMCNT output pin, page 91.

Public Variables

Public Variable Description

Variable	Type	Description
SC	MotorMNTGP_SC_DDT	Provides the frequently needed data to monitor the DFB status from the sequential control.

MotorMNTGP_SC_DDT Type

Name	Type	Description
RHCNT	UINT	Refer to the RHCNT output pin, page 91.
SHCNT	UINT	Refer to the SHCNT input/output pin, page 95.
PRHCNT	UINT	Refer to the PRHCNT input/output pin, page 95.
SRHCNT	UINT	Refer to the SRHCNT input/output pin, page 95.
MCNT	UINT	Refer to the MCNT output pin, page 91.
PMCNT	UINT	Refer to the PMCNT input/output pin, page 95.
SMCNT	UINT	Refer to the SMCNT input/output pin, page 95.
FSCNT	UINT	Refer to the FSCNT input/output pin, page 95.
ResetDays	UINT	Refer to the ResetDays input/output pin, page 95.

Name	Type	Description
TRHCNT	DINT	Refer to the TRHCNT output pin, page 91.
TMCNT	DINT	Refer to the TMCNT output pin, page 91.

Supervision - Alternate Services (Advance Motor Maintenance)

Overview

This section describes the Supervision resources and runtime services that are available for the advance motor maintenance.

Items

Overview

This section describes the variables and alarms items of advance motor maintenance.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from MotMntStatusWord.			
AdvMotMntHoursOverflow	Hours count overflow status	On/Off	*Bit0*
AdvMotMntOperationsOverflow	Operations count overflow status		*Bit1*
AdvMotMntStopHoursOverflow	Hours count overflow status		*Bit2*
AdvMotMntPrimaryHoursOverflow	Primary hours count overflow status	On/Off	*Bit3*
AdvMotMntSecondaryHoursOverflow	Secondary hours count overflow status		*Bit4*
AdvMotMntPrimaryOperationsOverflow	Primary operations count overflow status		*Bit5*
AdvMotMntSecondaryOperationsOverflow	Secondary operations count overflow status		*Bit6*
AdvMotMntFailOverflow	Stopped with fault count overflow status		*Bit7*
AdvMotMntResetDaysOverflow	Reset days from last reset couter overflow status	–	*Bit8*
AdvMotMntTotalHoursOverflow	Total hours count overflow status	On/Off	*Bit9*
AdvMotMntTotalOperationsOverflow	Total operations count overflow status		*Bit10*
Bit items derived from MotMntConfigurationWord.			
AdvMotMntResetCounters	Reset the counters	On/Off	*Bit0*

Item name	Description	Enu-mera-tion	Address
AdvMotMntResetTo-talCounters	Reset the total counters		*Bit1*
IO Items			
MotMntStatusWord	Motor maintenance status word	–	MOTMNTGP_ST.STW
MotMntConfigura-tionWord	Motor maintenance configuration word		MOTMNTGP_ST.CFGW
AdvMotMntTotalRun-Hours	Total run hours		MOTMNTGP_CFG. TotalRunHours
AdvMotMntStarts	Starts		MOTMNTGP_CFG.Starts
AdvMotMntStopped-Hours	Stopped hours		MOTMNTGP_CFG. StoppedHours
AdvMotMntRunHours	Run Hours		MOTMNTGP_CFG.RunHours
AdvMotMntRunHour-sPrimary	Run hours primary		MOTMNTGP_CFG. RunHoursPrimary
AdvMotMntRunHours-Secondary	Run hours secondary		MOTMNTGP_CFG. RunHoursSecondary
AdvMotMntStartsPri-mary	Starts primary		MOTMNTGP_CFG. StartsPrimary
AdvMotMntStartsSe-condary	Starts secondary		MOTMNTGP_CFG. StartsSecondary
AdvMotMntFailCount	Fail count		MOTMNTGP_CFG. FailCount
AdvMotMntDaysReset-Maint	Days since last reset		MOTMNTGP_CFG. DaysResetMaint
AdvMotMntTotal-Starts	Total starts		MOTMNTGP_CFG. TotalStarts
AdvMotMntRunHours-WarnLimit	Run hours alert limit		MOTMNTGP_CFG. RunHoursWarnLimit
AdvMotMntRunHour-sAlmLimit	Run hours alarm limit		MOTMNTGP_CFG. RunHoursAlmLimit
AdvMotMntStartWarn-Limit	Starts alert limit		MOTMNTGP_CFG. StartWarnLimit
AdvMotMntStartAlm-Limit	Starts alarm limit		MOTMNTGP_CFG. StartAlmLimit
AdvMotMntFailWarn-Limit	Fails alert limit		MOTMNTGP_CFG. FailWarnLimit
AdvMotMntFailAlmLi-mit	Fails alarm limit		MOTMNTGP_CFG. FailAlmLimit
AdvMotMntDaysDue-WarnLimit	Days due alert limit		MOTMNTGP_CFG. DaysDueWarnLimit
AdvMotMntDays-DueAlmLimit	Days due alarm limit		MOTMNTGP_CFG. DaysDueAlmLimit

Alarms

The table describes the alarms that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
AdvMotMntOver-flow	Counter Overflow	MotorGP_2_MotMNTGP_STW <> 0
AdvMotMn-tRunHrsWarn	Run hours alert	MotorGP_2_MotMNTGP_ RUNHRS>MotorGP_2_MotMNTGP_ RunHrsWarnSP

Item name	Description	Expressions
AdvMotMn-tRunHrsAlm	Run hours alarm	MotorGP_2_MotMNTGP_RUNHRS>MotorGP_2_MotMNTGP_RunHrsAlmSP
AdvMotMnt-StartCntWarn	Starts count alert	MotorGP_2_MotMNTGP_STARTS>MotorGP_2_MotMNTGP_StartCntWarnSP
AdvMotMnt-StartCntAlm	Starts count Alarm	MotorGP_2_MotMNTGP_STARTS>MotorGP_2_MotMNTGP_StartCntAlmSP
AdvMotMnt-FailCntWarn	Failures count alert	MotorGP_2_MotMNTGP_FAILS>MotorGP_2_MotMNTGP_MFailCntWarnSP
AdvMotMnt-FailCntAlm	Failures count alarm	MotorGP_2_MotMNTGP_FAILS>MotorGP_2_MotMNTGP_MFailCntAlmSP
AdvMotMntDue-Warn	Maintenance required alert	MotorGP_2_MotMNTGP_RESETDAYS<MotorGP_2_MotMNTGP_MMaintDueWarnSP
AdvMotMnt-DueAlm	Maintenance required alarm	MotorGP_2_MotMNTGP_RESETDAYS<MotorGP_2_MotMNTGP_MMaintDueAlmSP

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

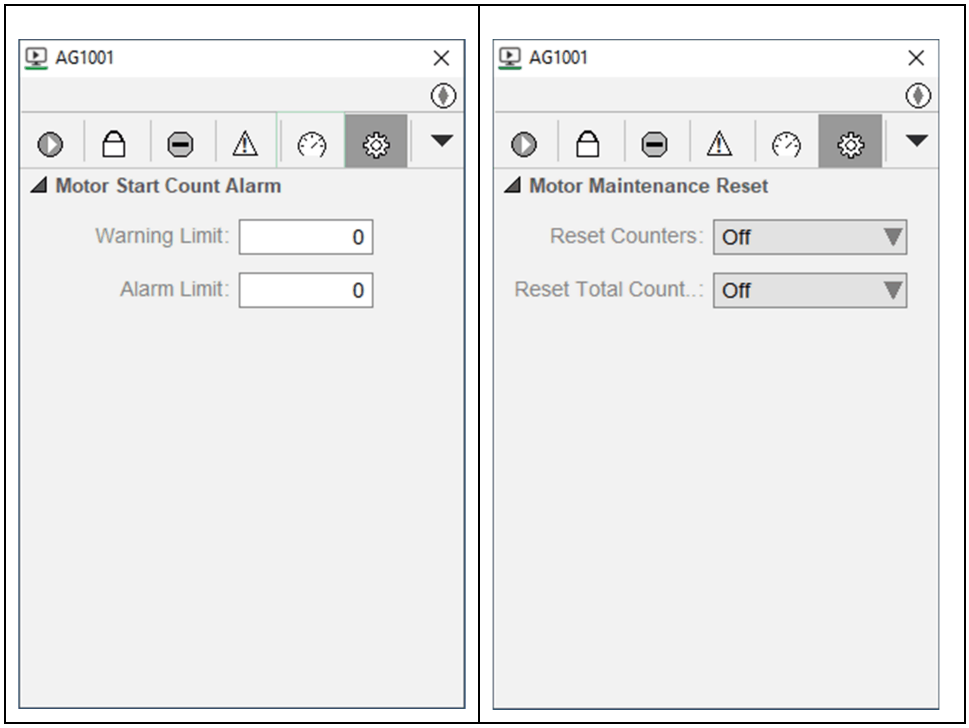
Available Tabs

During operation, clicking a motor genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Engineering
  - Measures

Engineering Tab

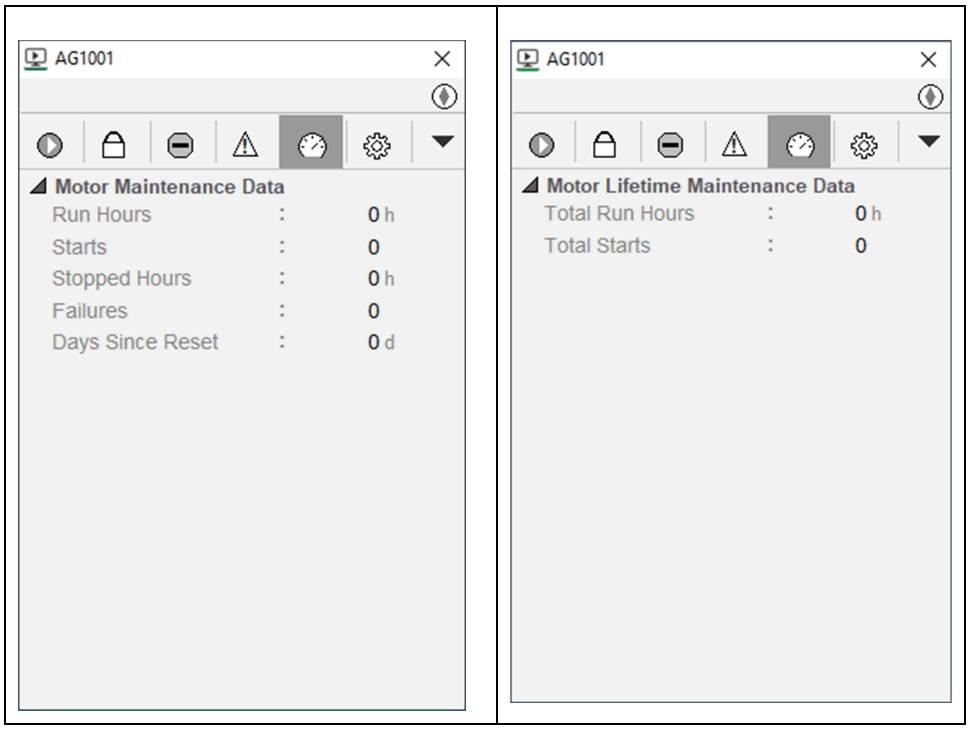
The figure shows an engineering tab:



For more details about maintenance group, refer to maintenance, page 92.

Measures Tab

The figure shows a measures tab:



Template - Alternate Services (Advance Valve Maintenance)

Maintenance

Description

The maintenance facet gathers run-hours and counts operations to enable maintenance systems. The maintenance values are available on the measures tab of the faceplate. The maintenance counters can be reset by clicking on the value displayed.

If the counters are not reset before the maximum value is reached a rollover alarm will be generated.

NOTE: In Engineering Tab, valve operation count alarm (`Warning Limit`, `Alarm Limit`) should be configured greater than zero to get valve operation count alarm and valve operation count warning.

Alternative maintenance facets supporting higher levels of maintenance functionality are available within the library.

Parameters

Supervision

The table describes the supervision parameters:

Section	Parameter	Type	Default	Description
Alarm	`Counter overflow of alarm severity`	*Enum*	Low (4000)	The default alarm severity is medium in all the templates.
	`Counter overflow of alarm group`	*String*	Maintenance	The default severity of all the maintenance alarm is low in all the templates.
Historize	`Enable historize counters`	*Boolean*	False	Enable historize counters.
	`Enable historize status/commands`			Enable historize status or commands.

Control - Alternate Services (Advance Valve Maintenance)

Description

General

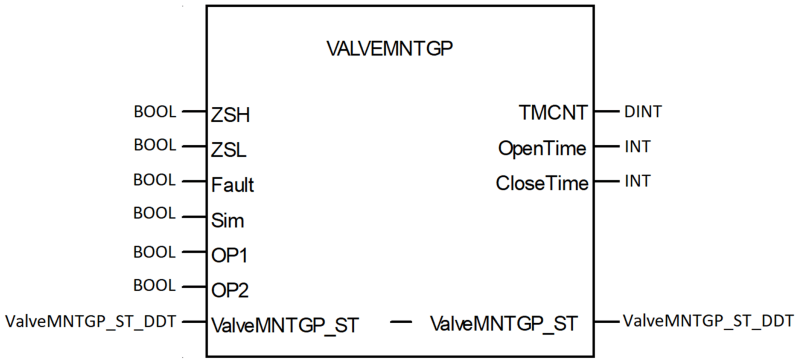
The `VALVEMNTGP` DFB is used to provide data that is useful for device maintenance.

You can use this DFB as a supplement to the `VALVEGP` DFB or independently. This DFB allows the incorporation of totalizing functions for hours of operation and switching operations.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
ZSH	BOOL	High position confirmation signal.
ZSL	BOOL	Low position confirmation signal.
Fault	BOOL	Device detected fault
Sim	BOOL	Enables to simulation mode.
OP1	BOOL	Field command signal for open position.
OP2	BOOL	Field command signal for close position.

Outputs

Output Parameter Description

Parameter	Type	Description
TMCNT	DINT	Total operations continuous counter.
OpenTime	IINT	Last total time to open (seconds).
CloseTime	IINT	Last total time to close (seconds).

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
ValveMNTG-P_ST	ValveMNTGP_ST_DDT	Data structure that is used as an interface with the monitoring subsystem.

ValveMNTGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the DFB status usually used from the monitoring subsystem. Read-only access to the data contained in this word.
	MOV	Bit0	BOOL	1 = Manuevers counter overflow bit.
	TMOV	Bit1	BOOL	1 = Total manuvers counter overflow bit (continous counters).
	FSOV	Bit2	BOOL	1 = Stopped with detected fault counter overflow bit.
	RDOV	Bit3	BOOL	1 = Reset days from last reset counter overflow bit.
	OPTOV	Bit5	BOOL	Time to open overflow bit.
	CLTOV	Bit6	BOOL	Time to close overflow bit.
CFGW	–	–	WORD	Provides the means to control the DFB. Read-only access to the data contained in this word.
	RSTCNT	Bit0	BOOL	1 = Reset bit for counters.
	RSTTOT	Bit1	BOOL	1 = Reset bit for total counters.
MCNT	–	–	UINT	The number of times a valve has opened or closed.
FSCNT	–	–	UINT	The number of times the valve lost the command with detected fault.
ResetDays	–	–	UINT	The number of days since last reset.
OpenTime	–	–	INT	Refer to the OpenTime output pin, page 102.
CloseTime	–	–	INT	Refer to the CloseTime output pin, page 102.
TMCNT	–	–	DINT	Refer to the TMCNT output pin, page 102.

Public Variables

Public Variable Description

Variable	Type	Description
SC	ValveMNTGP_SC_DDT	Provides the frequently needed data to monitor the DFB status from the sequential control.

ValveMNTGP_SC_DDT Type

Name	Type	Description
MCNT	UINT	Refer to the MCNT input/output pin, page 102.
FSCNT	UINT	Refer to the FSCNT input/output pin, page 102.
ResetDays	UINT	Refer to the ResetDays input/output pin, page 102.
OpenTime	INT	Refer to the OpenTime output pin, page 102.
CloseTime	INT	Refer to the CloseTime output pin, page 102.
TMCNT	DINT	Refer to the TMCNT output pin, page 102.

Supervision - Alternate Services (Advance Valve Maintenance)

Overview

This section describes the Supervision resources and runtime services that are available for the `advance valve maintenance`.

Items

Overview

This section describes the variables and alarms items of `advance valve maintenance`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from AdvVlvMntStatusWord.			
`AdvVlvMntOperationsOver-flow`	Operations count overflow status	On/Off	*Bit0*
`AdvVlvMntTotalOperation-sOverflow`	Total operations count overflow status		*Bit1*
`AdvVlvMntFailOverflow`	Lost command with fault count overflow status		*Bit2*
`AdvVlvMntDaysResetMaint`	Reset days since last reset	–	*Bit3*
`AdvVlvMntOpenTimeOver-flow`	Time to open overflow status		*Bit5*
`AdvVlvMntCloseTimeOver-flow`	Time to close overflow status		*Bit6*
Bit items derived from AdvVlvMntConfigurationWord.			
`AdvVlvMntResetCounters`	Reset the counters	On/Off	*Bit0*
`AdvVlvMntResetTotal-Counters`	Reset the total counters		*Bit1*
IO Items			
`AdvVlvMntStatusWord`	Valve maintenance status word	–	`VALVEMNTGP_ST.STW`
`AdvVlvMntConfiguration-Word`	Valve maintenance configuration word		`VALVEMNTGP_ST.CFGW`
`AdvVlvMntOperations`	Operations		`VALVEMNTGP_ST.STW`
`AdvVlvMntLastOpenTime`	Last open time		`VALVEMNTGP_CFG. LastOpenTime`
`AdvVlvMntLastCloseTime`	Last close time		`VALVEMNTGP_CFG. LastCloseTime`
`AdvVlvMntTotalOperations`	Total operations		`VALVEMNTGP_CFG. TotalOperations`
`AdvVlvMntFailCount`	Fail count	–	`VALVEMNTGP_CFG. FailCount`
`AdvVlvMntDaysResetMaint`	Days since last reset		`VALVEMNTGP_CFG. DaysResetMaint`
`AdvVlvMntOpenWarnTime`	Open alert time		`VALVEMNTGP_CFG. OpenWarnTime`

Item name	Description	Enumeration	Address
AdvVlvMntCloseWarnTime	Close alert time		VALVEMNTGP_CFG. CloseWarnTime
AdvVlvMntOperationsWarn-Limit	Operations alert limit		VALVEMNTGP_CFG. OperationsWarnLi-mit
AdvVlvMntOperationsAlm-Limit	Operations alarm limit		VALVEMNTGP_CFG. OperationsAlmLimit
AdvVlvMntFailWarnLimit	Fail alert limit		VALVEMNTGP_CFG. FailWarnLimit
AdvVlvMntFailAlmLimit	Fail alarm limit		VALVEMNTGP_CFG. FailAlmLimit
AdvVlvMntDaysDueWarnLi-mit	Days due alert limit		VALVEMNTGP_CFG. DaysDueWarnLimit
AdvVlvMntDaysDueAlmLimit	Days due alarm limit		VALVEMNTGP_CFG. DaysDueAlmLimit

Alarms

The table describes the alarms that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
AdvVlvMntOver-flow	Counter overflow	ValveGP_2_MNTGP_STW <> 0
AdvVlvMntO-perCntWarn	Starts count alert	ValveGP_2_MNTGP_ OPERATIONS>ValveGP_2_MNTGP_ OperCntWarnSP
AdvVlvMntO-perCntAlm	Starts count alarm	ValveGP_2_MNTGP_ OPERATIONS>ValveGP_2_MNTGP_ OperCntAlmSP
AdvVlvMnt-FailCntWarn	Failures count alert	ValveGP_2_MNTGP_ FAILS>ValveGP_2_MNTGP_ VFailCntWarnSP
AdvVlvMnt-FailCntAlm	Failures count alarm	ValveGP_2_MNTGP_ FAILS>ValveGP_2_MNTGP_ VFailCntAlmSP
AdvVlvMntDays-DueWarn	Maintenance required alert	ValveGP_2_MNTGP_ RESETDAYS<ValveGP_2_MNTGP_ VMaintDueWarnSP
AdvVlvMntDays-DueAlm	Maintenance required alarm	ValveGP_2_MNTGP_ RESETDAYS<ValveGP_2_MNTGP_ VMaintDueAlmSP
AdvVlvMntTra-velOpenWarn	Travel open alert	ValveGP_2_MNTGP_ LASTOPEN>ValveGP_2_MNTGP_ VOpenWarnTimeSP
AdvVlvMntTra-velCloseWarn	Travel close alert	ValveGP_2_MNTGP_ LASTCLOSE>ValveGP_2_MNTGP_ VCloseWarnTimeSP

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
ActionAlarm	All actions by the operator.
ModeAlarm	All changes in the mode and owner.
ActionAlarm	All actions by the operator.

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

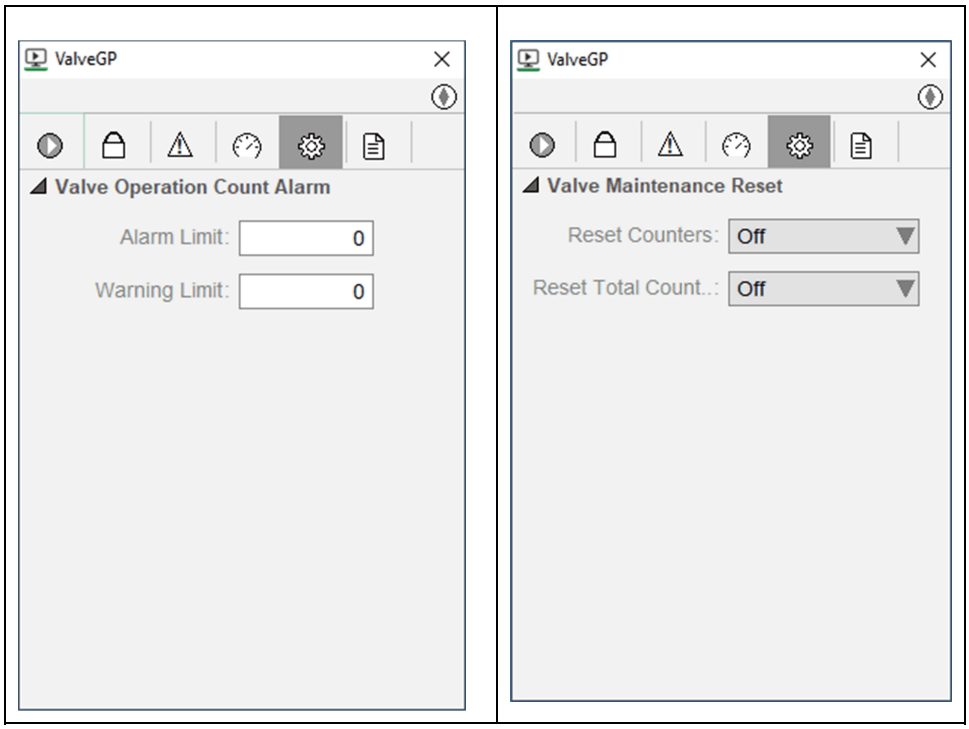
Available Tabs

During operation, clicking a valve genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Engineering
  - Measures

Engineering Tab

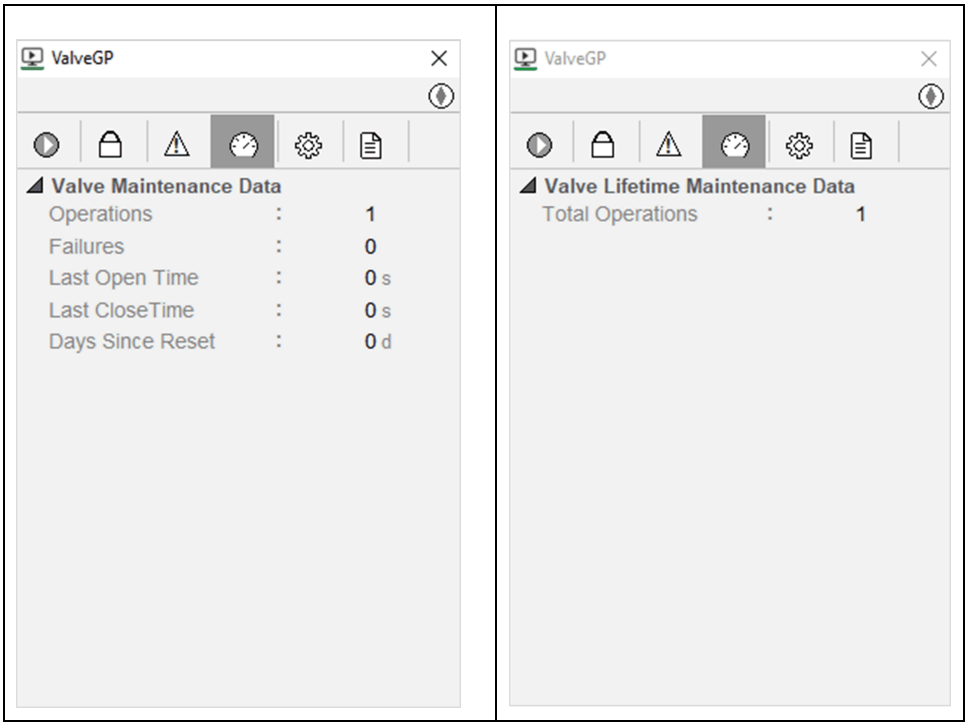
The figure shows an engineering tab:



For more details about maintenance group, refer to maintenance, page 92.

Measures Tab

The figure shows a measures tab:


ValveGP
Valve Maintenance Data
Operations : 1
Failures : 0
Last Open Time : 0 s
Last CloseTime : 0 s
Days Since Reset : 0 d
ValveGP
Valve Lifetime Maintenance Data
Total Operations : 1

Analog Alarm

What's in This Chapter

Template 108
Control 110
Supervision 114
Template - Alternate Service (LALARM) 118
Control - Alternate Service (LALARM) 120
Supervision - Alternate Services (LALARM) 123
Template - Alternate Service (DALARM) 124
Control - Alternate Service (DALARM) 126
Supervision - Alternate Services (DALARM) 128
Template - Alternate Service (RALARM) 130
Control - Alternate Service (RALARM) 131
Supervision - Alternate Services (RALARM) 133

Overview

This chapter describes the components that provide the functionality of template, control services, and Supervision functions of the Analog Alarm.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$AAlarmGP` template.

Description

General Description

The `AALARMGP` DFB is used to evaluate detected timed alarms associated with an analog signal.

The DFB provides detected alarm functions by level (very high, high, low and/or very low) and by deviation in relation to a set-point value.

This evaluation can be activated/deactivated individually according to the specific needs of the process both from the continuous and sequential control strategies implemented in the controller as well as from the monitoring system.

Function Description

The main functions of the DFB are described in the following table:

Function	Description
Level alarm	Evaluates detected timed alarms by level. The detected alarm connection can be timed. Timing and/or hysteresis can be applied to the disconnection.
Deviation alarm	Evaluates the detected alarm for maximum deviation in relation to a set-point.
Enabling	The DFB allows detected alarm monitoring to be enabled/disabled individually.

Parameters

Parameters

The `$AAlarmGP` template provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control alarm parameters:

Section	Parameter description	Type	Default	Additional remarks
Config-uration	Disconnection hysteresis	`Float`	0.0	–
	High high alarm	`Enum`	True Value (1)	Allows enabling or disabling the monitoring of the corresponding alarm. You can configure the input in one of the following ways: • **True Value**: Enables monitoring of the alarm by setting the value of the corresponding pin to TRUE. • **False**: Disables monitoring of the alarm by setting the value of the corresponding pin to FALSE. • **Interface**: Enables the corresponding interface, allowing you to provide the value through an interface of type $Bool. • **Refinement**: Allows you to set the value during refinement. The default value that is generated for the corresponding pin is NULL.
	High alarm			
	Low alarm			
	Low low alarm			
	Deviation alarm			
Time	Timing setpoint for connecting the alarms	`Dura-tion`	00.00.00	–
	Timing setpoint for disconnecting the alarms		00.00.00	–

Supervision

The table describes the supervision alarm parameters:

Section	Parameter description	Type	Default	Additional remarks
Alarm	High High description	`String`	@(High High Alarm)	–
	High High severity	`Enum`	Low (4000)	You can set the criticality (Critical (1000), High (2000), Medium (3000) and Low (4000)) of alarm.
	High High disable	`Boolean`	False	–
	High description	`String`	@(High Alarm)	
	High severity	`Enum`	Low (4000)	You can set the criticality (Critical (1000), High (2000),

Section	Parameter description	Type	Default	Additional remarks
				Medium (3000) and Low (4000)) of alarm.
	High disable	Boolean	False	–
	Low description	String	@(Low Alarm)	–
	Low severity	Enum	Low (4000)	You can set the criticality (Critical (1000), High (2000), Medium (3000) and Low (4000)) of alarm.
	Low disable	Boolean	False	–
	Low Low description	String	@(Low Low Alarm)	–
	Low Low severity	Enum	Low (4000)	You can set the criticality (Critical (1000), High (2000), Medium (3000) and Low (4000)) of alarm.
	Low Low disable	Boolean	False	–
	Deviation description	String	@(Deviation Alarm)	–
Alarm	Deviation setpoint severity	Enum	Low (4000)	–
	Deviation setpoint disable	Boolean	False	–
	Alarm group	String	Alarm	–
Events	Event log	Boolean	False	Enable event logging for asset status.
Trend	Setpoints and limits			–
Histor-ize	Setpoints and limits			–

Control

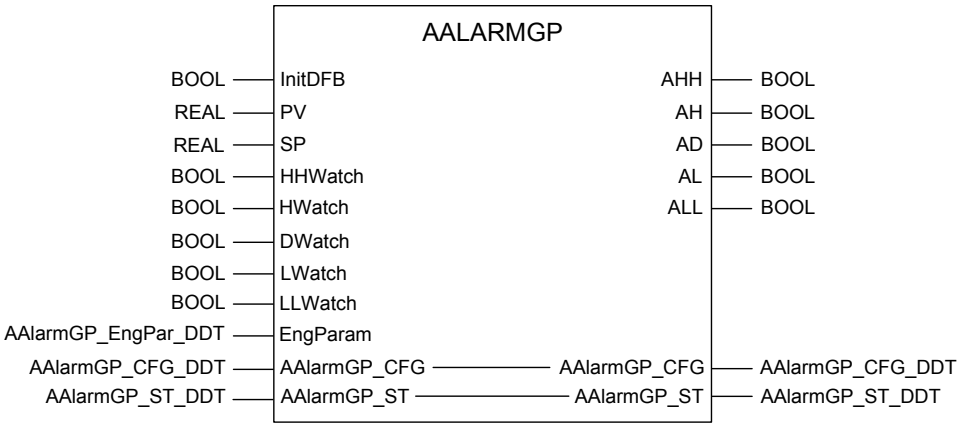
Overview

This section describes the AALARMGP DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
`InitDFB`	`BOOL`	1 = Indicates rising edge is detected on this input, the DFB internal timers are reset and detected alarms are set to 0.
`PV`	`REAL`	Value of the measurement with which the detected alarms are evaluated.
`SP`	`REAL`	Set-point level used as a reference for deviation alarm evaluation in engineering units (those associated with the corresponding `PV`)).
`HHWatch`	`BOOL`	1 = Allows very-high-level alarm monitoring to be enabled or disabled depending on the dynamic conditions of the process.
`HWatch`	`BOOL`	1 = Allows high-level alarm monitoring to be enabled or disabled depending on the dynamic conditions of the process.
`DWatch`	`BOOL`	1 = Allows dynamic alarm monitoring to be enabled (1) or disabled (0) depending on the dynamic conditions of the process.
`LWatch`	`BOOL`	1 = Allows low-level alarm monitoring to be enabled (1) or disabled (0) depending on the dynamic conditions of the process.
`LLWatch`	`BOOL`	1 = Allows very-low-level alarm monitoring to be enabled (1) or disabled (0) depending on the dynamic conditions of the process.
`EngParam`	`AAlarmGP_EngPar_DDT`	Engineering parameters

`AAlarmGP_EngPar_DDT` Type

Name	Type	Description
`OnDelay`	`TIME`	Timing set-point for connecting the detected alarms.
`OffDelay`	`TIME`	Timing set-point for disconnecting the detected alarms.
`HYSP`	`REAL`	Hysteresis set-point for disconnecting the detected alarms (in engineering units).

Outputs

Output Parameter Description

Parameter	Type	Description
AHH	BOOL	1 = Very-high-level alarm activated.
AH	BOOL	1 = High-level alarm activated.
AD	BOOL	1 = Deviation alarm activated.
AL	BOOL	1 = Low-level alarm activated.
ALL	BOOL	1 = Very-low-level alarm activated.

NOTE: Each detected alarm is timed independently according to the configuration of the OnDelay (for detected alarm activation) and OffDelay (for detected alarm deactivation) of engineering parameters. For disconnection purposes, the configured hysteresis is also considered.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
AAlarmGP_CFG	AAlarmGP_CFG_DDT	Analog alarm configuration parameters.
AAlarmGP_ST	AAlarmGP_ST_DDT	Provides the data necessary to monitor the detected alarms status.

AAlarmP_CFG Type

Name	Type	Description
SPHH	REAL	Very-high-level set-point in engineering units (those associated with the corresponding PV).
SPH	REAL	High-level set-point in engineering units (those associated with the corresponding PV).
SP	REAL	Setpoint level used as a reference for deviation alarm evaluation in engineering units (those associated with the corresponding PV).
SPD	REAL	Set-point for the maximum deviation allowed in the measurement in relation to the set-point configured in AAlarmP_CFG.SP in engineering units (those associated with the corresponding PV).
SPL	REAL	Low-level set-point in engineering units (those associated with the corresponding PV).
SPLL	REAL	Very-low-level set-point in engineering units (those associated with the corresponding PV).

AAlarmGP_ST_DDT Word Structure

This pin STW is connected to the STW field of the AAlarmGP_ST data structure of the related AAlarmGP variable in the event that the latter has been processed through an AAlarmGP-type DFB. This way, information generated from the AAlarmGP DFB is added to the information that the AAlarmGP DFB has generated in its status word.

Name	Bit Name	Bit	Type	Description
STW	–	–	*WORD*	Status word

Name	Bit Name	Bit	Type	Description
	HHWatch	0	*BOOL*	Refer to the HHWatch input pin, page 111.
	HWatch	1	*BOOL*	Refer to the HWatch input pin, page 111.
	DWatch	2	*BOOL*	Refer to the DWatch input pin, page 111.
	LWatch	3	*BOOL*	Refer to the LWatch input pin, page 111.
	LLWatch	4	*BOOL*	Refer to the LLWatch input pin, page 111.
	AHH	5	*BOOL*	Read-only access Refer to the AHH output pin, page 112.
	AH	6	*BOOL*	Read-only access Refer to the AH output pin, page 112.
	AD	7	*BOOL*	Read-only access Refer to the AD output pin, page 112.
	AL	8	*BOOL*	Read-only access Refer to the AL output pin, page 112.
	ALL	9	*BOOL*	Read-only access Refer to the ALL output pin, page 112.
CFGW	–	–	*WORD*	Configuration word
	EnHH	0	*BOOL*	Enables very-high-level alarm evaluation.
	EnH	1	*BOOL*	Enables high-level alarm evaluation.
	EnD	2	*BOOL*	Enables deviation alarm evaluation.
	EnL	3	*BOOL*	Enables low-level alarm evaluation.
	EnLL	4	*BOOL*	Enables very-low-level alarm evaluation.

Public Variables

Public Variable Description

Variable	Type	Description
SC	AAlarmP_SC_DDT	Provides the frequently needed data to monitor the status of the detected alarms from the sequential control.

AAlarmP_SC_DDT Type

Name	Type	Description
AHH	BOOL	Read-only access. Refer to the AHH output pin, page 112.
AH	BOOL	Read-only access. Refer to the AH output pin, page 112.
AD	BOOL	Read-only access. Refer to the AD output pin, page 112.
AL	BOOL	Read-only access. Refer to the AL output pin, page 112.
ALL	BOOL	Read-only access. Refer to the ALL output pin, page 112.

Supervision

Overview

This section describes the Supervision functions and runtime services that are available for the `Analog Alarm`.

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a general description of the graphic elements that are used in the faceplate.

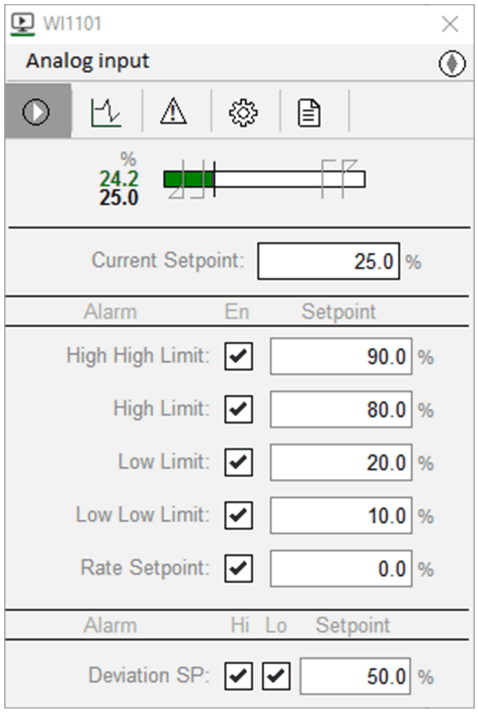
Available Tabs

During operation, clicking an analog input genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
  - Event, page 69
- Tabs for optional functions, which appear only if configured:
  - Measures, page 70

Operator Tab

The figure shows an operator tab:



Following actions can be performed from an operator tab:

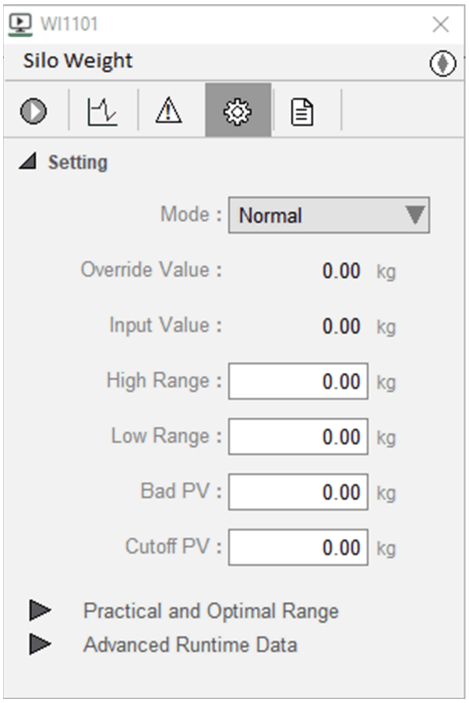
- **En**: Allows you to enable or disable the evaluation of an alarm conditions at the controller level.
- **Setpoint**: Allows you to set the setpoint.

The table describes the control function of an operator tab:

Label	Item name	Description	Security parameter, page 380
Current Setpoint	SP	Reference setpoint	Set setpoint
High High Limit	HighHighLimit, HighHighEnable	Very-high-level setpoint	
High Limit	HighLimit, HighEnable	High-level setpoint	
Low Limit	LowLimit, LowEnable	Low-level setpoint	
Low Low Limit	LowLowLimit, Add LowLowEnable Item	Very-low-level setpoint	
Rate Setpoint	RateEnbale, RateSetpoint	Rate setpoint	
Deviation SP	SPDeviationLimit, HighDeviationEnable, LowDeviationEnable	Deviation limit setpoint	

Engineering Tab

The figure shows an engineering tab:



The table describes the control function of an engineering tab:

Label	Item name	Description	Security parameter, page 380
Mode and Range			
Mode	Mode	Allows you to view and change the mode. Refer to operating mode, page 59.	Set mode - **override**, set mode - **maintenance**.
Override Value	OverrideValue	If **override** is selected, then input value is equal to	Set mode - override

Label	Item name	Description	Security parameter, page 380
		external PV, else input value is equal to channel value.	
Input Value	InputValue	Input value shows the current channel value.	–
High Range	PVHigh	Configurable high range.	Set parameter
Low Range	PVLow	Configurable low range.	Set parameter
Bad PV	PVBad	Value to consider when channel inoperable.	Set parameter
Cutoff PV	PVCutOff	Minimum reliable measure.	Set setpoint
Practical and Optimal Range			
Practical Range High	PRHigh	Practical range high	–
Practical Range Low	PRLow	Practical range low	–
Practical Range	PR	Practical range	Set setpoint
Optimal Range High	ORHigh	Optimal range high	–
Optimal Range Low	ORLow	Optimal range low	–
Advance Runtime Data			
High High Delay	HighHighLimitDelay	Delay high high	Set parameters
High High Hys	HighHighLimitHystere-sis	Hysteresis high high	
High Delay	HighLimitDelay	Delay high	
High Hys	HighLimitHysteresis	Hysteresis high	
Low Delay	LowLimitDelay	Delay low	
Low Hys	LowLimitHysteresis	Hysteresis low	
Low Low Delay	LowLowLimitDelay	Delay low low	

Items

Overview

This section describes the variables, abnormal and action required conditions, alarms, events, and trends items of Analog Alarm.

Variables

The table describes the variable items that are used by supervision components:

Item name	Description	Enumeration	Address
Bit items derived from AAlmStatusWord.			
`HighHighLimitWatch`	High-high alarm watch	–	*Bit0*
`HighLimitWatch`	High alarm watch		*Bit1*
`DeviationWatch`	Deviation alarm watch		*Bit2*
`LowLimitWatch`	Low alarm watch		*Bit3*
`LowLowLimitWatch`	Low-low alarm watch		*Bit4*

Item name	Description	Enumeration	Address
HighHighLimitAlarm	High-high alarm		*Bit5*
HighLimitAlarm	High alarm		*Bit6*
DeviationAlarm	Deviation alarm		*Bit7*
LowLimitAlarm	Low alarm		*Bit8*
LowLowLimitAlarm	Low-low alarm		*Bit9*
Bit items derived from AAlmConfigurationWord.			
HighHighEnable	High-high alarm enable	Enabled/Disabled	*Bit0*
HighEnable	High alarm enable		*Bit1*
DeviationEnable	Deviation alarm enable		*Bit2*
LowEnable	Low alarm enable		*Bit3*
LowLowEnable	Low-low alarm enable		*Bit4*
IO Items			
AAlmStatusWord	Aalarm status word		AALARMGP_ST.STW
AAlmConfigurationWord	Aalarm configuration word		AALARMGP_ST.CFGW

Alarms

The table describes the alarms that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
HighHighLimi-tAlarm	High-high alarm	GPL_BitValueCheck (StatusWord, 5)
HighLimitAlarm	High alarm	GPL_BitValueCheck (StatusWord, 6)
DeviationAlarm	Deviation alarm	GPL_BitValueCheck (StatusWord, 7)
LowAlarm	Low alarm	GPL_BitValueCheck (StatusWord, 8
LowLowLimitA-larm	Low-low alarm	GPL_BitValueCheck (StatusWord, 9)

Trends

The table describes the trend items that are used by Supervision components:

Item name	Expression	Description
SP	AALARMGP_CFG.SP	Setpoint
HighLimit	AALARMGP_CFG.SPH	High level setpoint
HighHighLi-mit	AALARMGP_CFG.SPHH	Very high level setpoint
LowLimit	AALARMGP_CFG.SPL	Low level setpoint
LowLowLimit	AALARMGP_CFG.SPLL	Very low level setpoint
DeviationLi-mit	AALARMGP_CFG.SPD	Deviation setpoint

Template - Alternate Service (LALARM)

Overview

This section describes functionality of the limit alarm template.

Description

General

The `LALARMGP` DFB is used to evaluate detected timed alarms associated with an analog signal.

The DFB provides detected alarm functions by level (very high, high, low and/or very low) and by deviation in relation to a set-point value.

This evaluation can be activated/deactivated individually according to the specific needs of the process both from the continuous and sequential control strategies implemented in the controller as well as from the monitoring system.

Function Description

The main functions of the DFB are described in the following table:

Function	Description
Level alarm	Evaluates detected timed alarms by level. The detected alarm connection can be timed. Timing and/or hysteresis can be applied to the disconnection.

Parameters

Parameters

The `$ALARMGP` template provides different limit alarm control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters of `LALARM`:

Section	Parameter description	Type	Default	Additional remarks
LimitAlarms				
Watch	High-High alarm	`Enum`	True value (1)	Allows enabling or disabling the monitoring of the corresponding alarm. You can configure the input in one of the following ways: • **True Value**: Enables monitoring of the alarm by setting the value of the corresponding pin to TRUE. • **False**: Disables monitoring of the alarm by setting the value of the corresponding pin to FALSE.
	High alarm			
	Low alarm			
	Low-Low alarm			

Section	Parameter description	Type	Default	Additional remarks
				• **Interface**: Enables the corresponding interface, allowing you to provide the value through an interface of type $Bool. • **Refinement**: Allows you to set the value during refinement. The default value that is generated for the corresponding pin is NULL.

Supervision

Data

The table describes the data parameters of `LALARM`:

Section	Parameter description	Type	Default	Additional remarks
LimitAlarms				
Description	High-High alarm	`String`	@ (High-High alarm)	–
	High alarm		@ (High alarm)	–
	Low alarm		@ (Low alarm)	–
	Low-Low alarm		@ (Low-Low alarm)	–
Alarm	High - High severity	`Enum`	Low (4000)	You can set the criticality (Critical (1000), High (2000), Medium (3000) and Low (4000)) of alarm.
	High - High alarm disable	`Boolean`	False	–
	High severity	`Enum`	Low (4000)	You can set the criticality (Critical (1000), High (2000), Medium (3000) and Low (4000)) of alarm.
	High alarm disable	`Boolean`	False	–
	Low severity	`Enum`	Low (4000)	You can set the criticality (Critical (1000), High (2000), Medium (3000) and Low (4000)) of alarm.
	Low alarm disable	`Boolean`	False	–
	Low-low severity	`Enum`	Low (4000)	You can set the criticality (Critical (1000), High (2000), Medium (3000) and Low (4000)) of alarm.
	Low-low alarm disable	`Boolean`	False	–
	Alarm group	`String`	Alarm	–
Trend	Disable limits	`Boolean`	False	–
Events	Enable log	`Boolean`	False	Enable event logging for asset status.
Historize	Enable historize status/commands	`Boolean`	False	–
	Enable historize setpoints			–

Control - Alternate Service (LALARM)

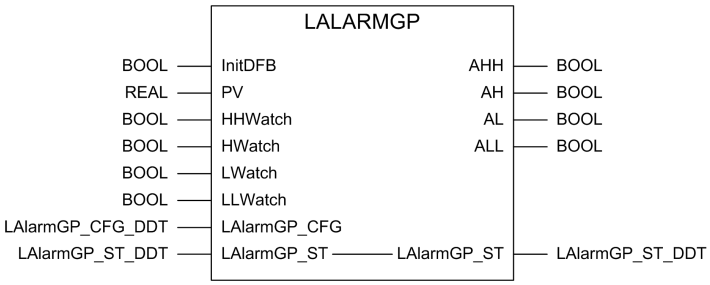
Overview

This section describes the `LALARMGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Name	Type	Description
`InitDFB`	`BOOL`	1 = Indicates rising edge is detected on this input, the DFB internal timers are reset and detected alarms are set to 0.
`PV`	`REAL`	Value of the measurement with which the detected alarms are evaluated.
`HHWatch`	`BOOL`	1 = Allows very-high-level alarm monitoring to be enabled or disabled depending on the dynamic conditions of the process.
`HWatch`	`BOOL`	1 = Allows high-level alarm monitoring to be enabled or disabled (0) depending on the dynamic conditions of the process.
`LWatch`	`BOOL`	1 = Allows low-level alarm monitoring to be enabled (1) or disabled (0) depending on the dynamic conditions of the process.
`LLWatch`	`BOOL`	1 = Allows very-low-level alarm monitoring to be enabled (1) or disabled (0) depending on the dynamic conditions of the process.
`LAlarmGP_CFG`	`LAlarmGP_CFG_DDT`	Configuration data necessary for evaluating the detected alarms. This data can be defined from the monitoring subsystem or from the control program.

`LAlarmGP_CFG_DDT` Type

Name	Type	Description
`SPHH`	`REAL`	Very-high-level set-point in engineering units associated with the corresponding `PV`.
`SPH`	`REAL`	High-level set-point in engineering units associated with the corresponding `PV`.

Name	Type	Description
SPL	REAL	Low-level set-point in engineering units associated with the corresponding PV.
SPLL	REAL	Very-low-level set-point in engineering units associated with the corresponding PV.
HYHHSP	REAL	Hysteresis set-point in engineering units associated with the corresponding SPHH.
HYHSP	REAL	Hysteresis set-point in engineering units associated with the corresponding SPH.
HYLSP	REAL	Hysteresis set-point in engineering units associated with the corresponding SPL.
HYLLSP	REAL	Hysteresis set-point in engineering units associated with the corresponding SPLL.
DelayHH	UINT	Delay time in seconds for enabling high-high alarm.
DelayH	UINT	Delay time in seconds for enabling high alarm.
DelayL	UINT	Delay time in seconds for enabling low alarm.
DelayLL	UINT	Delay time in seconds for enabling low-low alarm.

Outputs

Output Parameter Description

Name	Type	Description
AHH	BOOL	1 = Very-high-level alarm activated.
AH	BOOL	1 = High-level alarm activated.
AL	BOOL	1 = Low-level alarm activated.
ALL	BOOL	1 = Very-low-level alarm activated.

NOTE: Each detected alarm is timed independently according to the configuration of the TONSP (for detected alarm activation) and TOFFSP (for detected alarm deactivation) inputs. For disconnection purposes, the configured hysteresis is also considered.

Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
LAlarmGP_ST	LAlarmGP_ST_DDT	Provides the data necessary to monitor the detected alarms status.

LAlarmGP_ST_DDT Word Structure

This pin STW is connected to the STW field of the LAlarmGP_ST data structure of the related LALARMGP variable in the event that the latter has been processed through an LALARMGP-type DFB. This way, information generated from the LALARMGP DFB is added to the information that the LALARMGP DFB has generated in its status word. As a result, the diagnosis status (generated from the LALARMGP DFB) and you can include detected alarms status (generated from the LALARMGP DFB) in the same word.

Name	Bit Name	Bit	Type	Description
STW	–	–	*WORD*	Status word
	HHWatch	0	*BOOL*	Refer to the HHWatch input pin, page 120.
	HWatch	1	*BOOL*	Refer to the HWatch input pin, page 120.
	LWatch	2	*BOOL*	Refer to the LWatch input pin, page 120.
	LLWatch	3	*BOOL*	Refer to the LLWatch input pin, page 120.
	AHH	4	*BOOL*	Read-only access Refer to the AHH output pin, page 121.
	AH	5	*BOOL*	Read-only access Refer to the AH output pin, page 121.
	AL	6	*BOOL*	Read-only access Refer to the AL output pin, page 121.
	ALL	7	*BOOL*	Read-only access Refer to the ALL output pin, page 121.
CFGW	–	–	*WORD*	Configuration word
	EnHH	0	*BOOL*	Enables very-high-level alarm evaluation.
	EnH	1	*BOOL*	Enables high-level alarm evaluation.
	EnL	2	*BOOL*	Enables low-level alarm evaluation.
	EnLL	3	*BOOL*	Enables very-low-level alarm evaluation.

Public Variables

Public Variable Description

Variable	Type	Description
SC	LAlarmGP_SC_DDT	Provides the frequently needed data to monitor the sequence status and control it from the sequential control.

LAlarmGP_SC_DDT Type

Name	Type	Description
AHH	BOOL	Read-only access Refer to the AHH output pin, page 121.
AH	BOOL	Read-only access Refer to the AH output pin, page 121.
AL	BOOL	Read-only access Refer to the AL output pin, page 121.
ALL	BOOL	Read-only access Refer to the ALL output pin, page 121.

Supervision - Alternate Services (LALARM)

Overview

This section describes the Supervision functions and runtime services that are available for the analog alarm.

Items

Overview

This section describes the variables, alarms, and trends items of limit alarm.

Variables

The table describes the variable items that are used by supervision components:

Item name	Description	Enumeration	Address
Bit items derived from LAlmStatusWord.			
`HighHighLimitWatch`	High-high alarm watch	–	*Bit0*
`HighLimitWatch`	High alarm watch		*Bit1*
`LowLimitWatch`	Low alarm watch		*Bit2*
`LowLowLimitWatch`	Low-low alarm watch		*Bit3*
`HighHighLimitAlarm`	High-high alarm		*Bit4*
`HighLimitAlarm`	High alarm		*Bit5*
`LowLimitAlarm`	Low alarm		*Bit6*
`LowLowLimitAlarm`	Low-low alarm		*Bit7*
Bit items derived from LAlmConfigurationWord.			
`HighHighEnable`	High-high alarm enable	Enabled/Disabled	*Bit0*
`HighEnable`	High alarm enable		*Bit1*
`LowEnable`	Low alarm enable		*Bit2*
`LowLowEnable`	Low-low alarm enable		*Bit3*
IO Items			
`LAlmStatusWord`	Limit alarm status word	–	`LALRMGP_ST.STW`
`LAlmConfigurationWord`	Limit alarm configuration word	–	`LALRMGP_ST.CFGW`

Alarms

The table describes the alarms that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expression
`HighHighLimi-tAlarm`	High-high alarm	`GPL_BitValueCheck (StatusWord, 4)`
`HighLimitAlarm`	High alarm	`GPL_BitValueCheck (StatusWord, 5)`

Item name	Description	Expression
LowAlarm	Low alarm	GPL_BitValueCheck (StatusWord, 6)
LowLowLimitA-larm	Low-low alarm	GPL_BitValueCheck (StatusWord, 7)

Trends

The table describes the trend items that are used by Supervision components:

Item name	Expression	Description
HighLimit	LALARMGP_CFG.SPH	High level setpoint
HighHighLi-mit	LALARMGP_CFG.SPHH	Very high level setpoint
LowLimit	LALARMGP_CFG.SPL	low level setpoint
LowLowLimit	LALARMGP_CFG.SPLL	Very low level setpoint

Template - Alternate Service (DALARM)

Overview

This section describes functionality of the deviate alarm template.

Description

General

The DALARMGP DFB is used to evaluate detected timed alarms associated with an analog signal.

The DFB provides detected alarm functions by level (very high, high, low and/or very low) and by deviation in relation to a set-point value.

This evaluation can be activated/deactivated individually according to the specific needs of the process both from the continuous and sequential control strategies implemented in the controller as well as from the monitoring system.

Function Description

The main functions of the DFB are described in the following table:

Function	Description
Deviation alarm	Evaluates the detected alarm for maximum deviation in relation to a set-point.

Parameters

Parameters

The $ALARMGP template provides different deviation alarm control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters of DALARM:

Section	Parameter description	Type	Default	Additional remarks
DeviationAlarms				
Watch	Deviation alarm	Enum	True value (1)	For description, refer to **Watch** in LimitAlarms.
Setpoint	Deviation setpoint calculation	Enum	Abso-lute value (0)	Calculation of deviation setpoint based on absolute value or in percentage.
	Deviation limit setpoint type			Type of deviation signal to be considered. Example: Deviation setpoint or high/low deviation setpoint.

Supervision

Data

The table describes the data parameters of DALARM:

Section	Parameter description	Type	Default	Additional remarks
DeviationAlarms				
Configura-tion	Deviation practical range disabled	Boolean	True	**NOTE:** This parameter is applicable only for analog input.
Description	Deviation	String	@ (Deviation alarm)	–
	High deviation		@ (High deviation alarm)	–
	Low deviation		@ (Low deviation alarm)	–
Alarm	Deviation severity	Enum	Low (4000)	You can set the criticality (Critical (1000), High (2000), Medium (3000) and Low (4000)) of alarm.
	High deviation disable	Boolean	False	–
	High deviation severity	Enum	Low (4000)	You can set the criticality (Critical (1000), High (2000), Medium (3000) and Low (4000)) of alarm.
	Low deviation disable	Boolean	False	–
	Low deviation severity	Enum	Low (4000)	You can set the criticality (Critical (1000), High (2000), Medium (3000) and Low (4000)) of alarm.
	Deviation group	String	Deviation	–
Trend	Disable limits	Boolean	False	–
Events	Enable log	Boolean	False	–
Historize	Setpoints	Boolean	False	–

Control - Alternate Service (DALARM)

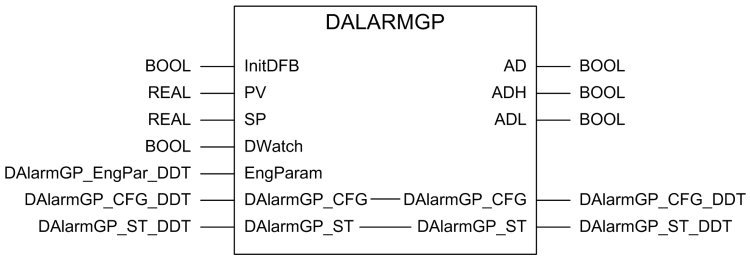
Overview

This section describes the `DALARMGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Name	Type	Description
`InitDFB`	`BOOL`	1 = Indicates rising edge is detected on this input, the DFB internal timers are reset and detected alarms are set to 0.
`PV`	`REAL`	Value of the measurement with which the detected alarms are evaluated.
`SP`	`REAL`	Set-point level used as a reference for deviation alarm evaluation in engineering units (those associated with the corresponding PV)).
`DWatch`	`BOOL`	1 = Allows dynamic alarm monitoring to be enabled (1) or disabled (0) depending on the dynamic conditions of the process.
`EngParam`	`DAlarmGP_EngPar_DDT`	Engineering parameters

`DAlarmGP_EngPar_DDT` Type

Name	Type	Description
`DLimSel`	`BOOL`	Deviation limit selection (1 = High and Low limit selected, 0 = Absolute limit selected)
`DPSel`	`BOOL`	Deviation setpoint in percentage (1 = In Percentage, 0 = Absolute)

Outputs

Output Parameter Description

Name	Type	Description
AD	BOOL	1 = Deviation alarm activated
ADH	BOOL	1 = High deviation alarm activated
ADL	BOOL	1 = Low deviation alarm activated

Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
DAlarmGP_CFG	DAlarmGP_CFG_DDT	Deviation alarm configuration parameters
DAlarmGP_ST	DAlarmGP_ST_DDT	Provides the data necessary to monitor the detected alarms status

DAlarmGP_CFG_DDT Type

Name	Type	Description
SP	REAL	Refer to SP input pin, page 126.
SPD	REAL	Set-point for the maximum deviation allowed in the measurement in relation to the set-point configured in DAlarmGP_CFG.SP in engineering units (those associated with the corresponding PV).
SPH	REAL	High-level set-point in engineering units (those associated with the corresponding PV).
SPL	REAL	Low-level set-point in engineering units (those associated with the corresponding PV).
HYSP	REAL	Hysteresis set-point in engineering units associated with the corresponding SP.
HYHSP	REAL	Hysteresis set-point in engineering units associated with the corresponding SPH.
HYLSP	REAL	Hysteresis set-point in engineering units associated with the corresponding SPL.
DelayDevia-tion	UINT	Delay time in seconds for enabling deviation.

DAlarmGP_ST_DDT Word Structure

This pin STW is connected to the STW field of the DAlarmGP_ST data structure of the related DALARMGP variable in the event that the latter has been processed through an DALARMGP-type DFB. This way, information generated from the DALARMGP DFB is added to the information that the DALARMGP DFB has generated in its status word. As a result, the diagnosis status (generated from the DALARMGP DFB) and you can include detected alarms status (generated from the DALARMGP DFB) in the same word.

Name	Bit Name	Bit	Type	Description
STW	–	–	*WORD*	Status word
	DWatch	Bit0	*BOOL*	Indicates if the deviation alarm is being monitored. Reproduces the value of the DWatch input pin.

Name	Bit Name	Bit	Type	Description
	AD	Bit1	*BOOL*	Refer to AD output pin, page 127.
	ADH	Bit2	*BOOL*	Refer to ADH output pin, page 127.
	ADL	Bit3	*BOOL*	Refer to ADL output pin, page 127.
	SPLink	Bit4	*BOOL*	1 = External SP linked.
CFGW	–	–	*WORD*	Configuration word
	EnDH	Bit0	*BOOL*	Enables high-deviation alarm evaluation.
	EnDL	Bit1	*BOOL*	Enables low-deviation alarm evaluation.

Public Variables

Public Variable Description

Variable	Type	Description
SC	DAlarmGP_SC_DDT	Provides the frequently needed data to monitor the status of the detected alarms from the sequential control.

DAlarmGP_SC_DDT Type

Name	Type	Description
SP	*REAL*	Read-only access Refer to SP input pin, page 126.
AD	*BOOL*	Read-only access Refer to AD output pin, page 127.
ADH	*BOOL*	Read-only access Refer to ADH output pin, page 127.
ADL	*BOOL*	Read-only access Refer to ADL output pin, page 127.

Supervision - Alternate Services (DALARM)

Overview

This section describes the Supervision functions and runtime services that are available for the analog alarm.

Items

Overview

This section describes the variables, alarms, and trends items of deviation alarm.

Variables

The table describes the variable items that are used by supervision components:

Item name	Description	Enumeration	Address
Bit items derived from DAlmStatusWord.			
DeviationWatch	Deviation alarm watch enable	–	*Bit0*
DeviationAlarm	Deviation alarm active		*Bit1*
HighDeviationAlarm	High deviation alarm active		*Bit2*
LowDeviationAlarm	Low deviation alarm active		*Bit3*
SPLink	Setpoint deviation alarm from program		*Bit4*
Bit items derived from DAlmConfigurationWord.			
HighDeviationEnable	High deviation alarm enable	Enable/Disable	*Bit0*
LowDeviationEnable	Low deviation alarm enable		*Bit1*
IO Items			
SP	Setpoint	–	DALRMGP_CFG.SP
DeviationHysteresis	Hysteresis value for deviation alarm		DALRMGP_CFG.HYSP
DeviationDelay	Delay for enabling deviation		DALRMGP_CFG. DELAYDEVIATION
LowDeviationHysteresis	Low deviation Hys		DALRMGP_CFG.HYLSP
DeviationLowSetpoint	Low deviation SP		DALRMGP_CFG.SPL
DeviationHighSetpoint	High deviation SP		DALRMGP_CFG.SPH
HighDeviationHysteresis	High deviation Hys		DALRMGP_CFG.HYHSP
DAlmStatusWord	Status word		DALRMGP_ST.STW
DAlmConfigurationWord	Configuration word		DALRMGP_ST.CFGW
DeviationSetpoint	Deviation SP		DALRMGP_CFG.SPD

Alarms

The table describes the alarms that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
HighDeviatio-nAlarm	High deviation alarm active	GPL_BitValueCheck (StatusWord, 2)
LowDeviationA-larm	Low deviation alarm active	GPL_BitValueCheck (StatusWord, 3)

Trends

The table describes the trend items that are used by Supervision components:

Item name	Expression	Description
HighDevia-tion	DALARMGP_CFG.SPH	High deviation
LowDeviation	DALARMGP_CFG.SPL	Low deviation

Template - Alternate Service (RALARM)

Overview

This section describes functionality of the rate alarm template.

Description

General

The RALARMGP DFB is used to evaluate detected rate alarms associated with an analog signal.

The DFB provides detected alarm functions by level (very high, high, low and/or very low) and by deviation in relation to a set-point value.

This evaluation can be activated/deactivated individually according to the specific needs of the process both from the continuous and sequential control strategies implemented in the controller as well as from the monitoring system.

Function Description

The main functions of the DFB are described in the following table:

Function	Description
RALARMGP	The DFB allows detected rate alarm monitoring to be enabled/disabled individually.

Parameters

Parameters

The $ALARMGP template provides different rate alarm control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters of RALARM:

Section	Parameter description	Type	Default	Additional remarks
RateAlarms				
Watch	Rate Alarm	Enum	True value (1)	For description, refer to **Watch** in LimitAlarms.
Alarm	Minimum time to maintain alarm (second)	Integer	5	Minimum time to maintain the confirmation alarm or detected fail alarm.

Supervision

Data

The table describes the data parameters of RALARM:

Section	Parameter description	Type	Default	Additional remarks
RateAlarms				
Description	Rate alarm description	`String`	@ (Rate alarm)	–
Alarm	Rate severity	`Enum`	Low (4000)	You can set the criticality (Critical (1000), High (2000), Medium (3000) and Low (4000)) of alarm.
	Rate alarm disable	`Boolean`	False	–
	rate alarm group	`String`	Rate	–
Trend	Disable trend	`Boolean`	False	–
Events	Enable log	`Boolean`	False	–
Historize	Enable historize status/commands	`Boolean`	False	It allows to historize the change in status or commands.
	Enable historize setpoint			It allows to historize the change in setpoint.

Control - Alternate Service (RALARM)

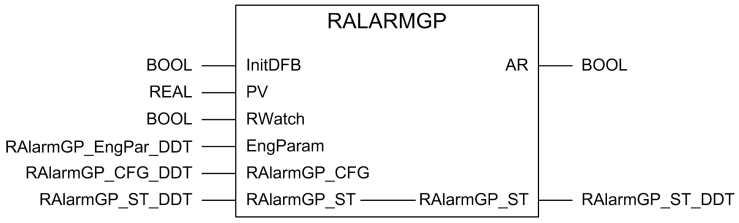
Overview

This section describes the `RALARMGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Name	Type	Description
`InitDFB`	`BOOL`	1 = Indicates rising edge is detected on this input, the DFB internal timers are reset and detected alarms are set to 0.
`PV`	`REAL`	Value of the measurement with which the detected alarms are evaluated.
`RWatch`	`BOOL`	Rate alarm monitoring (1 = Enable, 0 = Disable).

Name	Type	Description
EngParam	RAlarmGP_EngPar_DDT	Engineering parameters
RAlarmGP_CFG	RAlarmGP_CFG_DDT	Configuration data necessary for evaluating the detected alarms. This data can be defined from the monitoring subsystem or from the control program.

RAlarmGP_EngPar_DDT Type

Name	Type	Description
ScanTime	*UNIT*	The minimum time for which the detected alarm signals are kept active. ScanTime confirms that the supervision system acquires the data for detected alarms that are automatically reset.

RAlarmGP_CFG_DDT Type

Name	Type	Description
SPRate	*REAL*	Rate difference (%)
SPRTime	*UINT*	Timer to push PV to OLDPV.

Outputs

Output Parameter Description

Name	Type	Description
AR	*BOOL*	1 = Rate alarm activated

Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
RAlarmGP_ST	RAlarmGP_ST_DDT	Provides the data necessary to monitor the detected alarms status.

RAlarmGP_ST_DDT Word Structure

This pin STW is connected to the STW field of the RAlarmGP_ST data structure of the related analog variable in the event that the latter has been processed through an RALARMGP-type DFB. This way, information generated from the RALARMGP DFB is added to the information that the RALARMGP DFB has generated in its status word. As a result, the diagnosis status (generated from the AINPUT DFB) and you can include detected alarms status (generated from the RALARMGP DFB) in the same word.

Name	Bit Name	B-it	Type	Description
STW	–	–	*WORD*	Status word
	RWatch	0	*BOOL*	Indicates if the rate alarm is being monitored. Reproduces the value of the RWatch input pin.

Name	Bit Name	B-it	Type	Description
	AR	1	*BOOL*	Refer to the AR ouput pin, page 132.
CFGW	–	–	*WORD*	Configuration word
	EnAR	0	*BOOL*	Enables rate alarm evaluation.

Public Variables

Public Variable Description

Variable	Type	Description
SC	RAlarmGP_SC_DDT	Provides the frequently needed data to monitor the sequence status and control it from the sequential control.

RAlarmGP_SC_DDT Type

Name	Type	Description
AR	BOOL	Read-only access Refer to the AR ouput pin, page 132.

Supervision - Alternate Services (RALARM)

Overview

This section describes the Supervision functions and runtime services that are available for the analog alarm.

Items

Overview

This section describes the variables, alarms, and trends items of rate alarm.

Variables

The table describes the variable items that are used by supervision components:

Item name	Description	Enumeration	Address
Bit items derived from RAlmStatusWord.			
RateWatch	Rate alarm watch enable	–	*Bit0*
RateAlarm	Rate alarm active		*Bit1*
Bit items derived from RAlmConfigurationWord.			
RateEnable	Rate enable	Active/Inactive	*Bit0*
IO Items			
RateScanTime	Scan time	–	RALARMGP_CFG.SCANTIME
RateTimebase	Time base rate setpoint		RALARMGP_CFG.SPRTIME

Item name	Description	Enumeration	Address
RateSetpoint	Setpoint rate (%)		RALARMGP_CFG.SPRATE
RAlmStatusWord	Status word	–	RALARMGP_ST.STW
RAlmConfigurationWord	Configuration word		RALARMGP_ST.CFGW

Alarms

The table describes the alarms that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
RateAlarm	Rate alarm	GPL_BitValueCheck (StatusWord, 1)

Trends

The table describes the trend items that are used by Supervision components:

Item name	Expression	Description
RateSetpoint	RALARMGP_CFG.SP	Rate setpoint

Group Control Link

The group data pin on the template allows the user to connect the asset to `$GroupMGTGP/$GroupCTLGP` template which are available in EcoStruxure process Expert Group Control library.

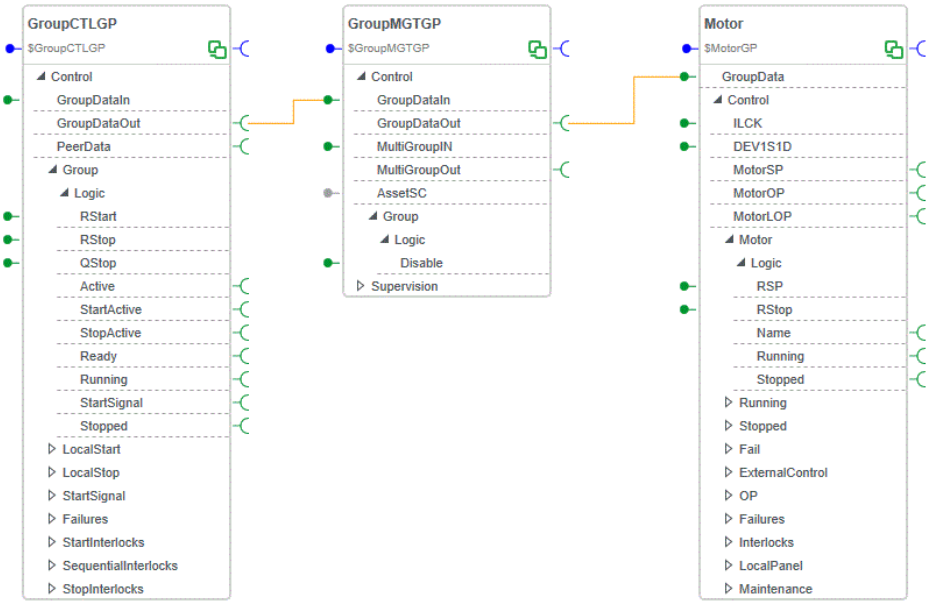
Refer to **EcoStruxure™ Process Expert - Group Control Library** for more details about group control library templates (`$GroupMGTGP, $GroupCTLGP, $GroupPCGP, and $GroupPOGP`).

The following table list the templates with group control link:

Name	Description
`$ControlValveGP`	Control Valve
`$MotorVSGP`	Motor Variable Speed
`$MValveWithPosGP`	Motorized Valve (with Positioner)
`$DualOPValveGP`	Dual Output Valve
`$HandValveGP`	Hand Valve
`$Motor2DirGP`	Motor Two Directions
`$Motor2SpdGP`	Motor Two Speed
`$MotorGP`	Motor One Direction
`$MValveGP`	Motorized Valve (without Positioner)
`$ValveGP`	Valve One Output
`$PIDGP`	PID
`$IMCGP`	Internal Model Controller
`$LeadLagGP`	Lead Lag
`$SplitRangeGP`	Split Range Controller
`$Step3CtlGP`	Step 3 Controller
`$PWMOutputGP`	Pulse Width Modulated Output
`$AnalogInMultiGP`	Multiple Analog Input
`$PhaseGP`	Phase Manager
`$TotalGP`	Total
`$SequenceGP`	Sequence Control
`$RatioCtrlGP`	Ratio Controller
`$AnalogInputGP`	Analog Indicator with Configuration Range
`$DigitalInputGP`	Digital Input
`$DigitalOutputGP`	Digital Output
`$EMPatternGP`	Equipment Module Pattern
`$PumpSetCtrlGP`	Pump Set Control Pattern
`$PumpFlowCtrlGP`	Pump Flow Control Pattern
`$PIDADVGP`	Advanced PID
`$PID2LoopGP`	Two loop PID control
`$PID4LoopGP`	Four loop PID control

Using Group Control Link

The following figure shows an example of group data pin connected to the `$GroupMGTGP` template:


GroupCTLGP
$GroupCTLGP
Control
GroupDataIn
GroupDataOut
PeerData
Group
Logic
RStart
RStop
QStop
Active
StartActive
StopActive
Ready
Running
StartSignal
Stopped
LocalStart
LocalStop
StartSignal
Failures
StartInterlocks
SequentialInterlocks
StopInterlocks
GroupMGTGP
$GroupMGTGP
Control
GroupDataIn
GroupDataOut
MultiGroupIN
MultiGroupOut
AssetSC
Group
Logic
Disable
Supervision
Motor
$MotorGP
GroupData
Control
ILCK
DEV1S1D
MotorSP
MotorOP
MotorLOP
Motor
Logic
RSP
RStop
Name
Running
Stopped
Running
Stopped
Fail
ExternalControl
OP
Failures
Interlocks
LocalPanel
Maintenance

Motors

What's in This Part

Motor 138
Dual Speed and Dual Direction Motor 162
Variable Speed Motor 197

Overview

This part provides the functionality of the templates, a detailed description of the control (functions, pins, pin layout, and variables of the function blocks), and the supervision functions of motors.

These components do not reflect any specific installation.

⚠ WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate them.
- Review the implications of communication link interruptions and take actions to mitigate them.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, and applicable codes and regulations.
- Apply local accident prevention and safety regulations and guidelines.[1]
- Test each implementation of a system for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

Motor

What's in This Chapter

Template 138
Control 143
Optional Service - Local Panel (MOTORLPGP) 150
Supervision..... 154

Overview

This chapter describes the functionality of template, control services, and supervision functions of the motor.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$MotorGP` template.

Description

General Description

The `$MotorGP` process control module template is used to manage one direction motor. The template allows managing the associated devices from the sequential control, the continuous/trigger based control, and/or the supervision system. It depends on the configuration of the associated devices and the system requirements.

Functional Description

The main functions of motor template are described in the following table:

Function	Description
Owner selection	The object manages the control system level, which is the owner (operator or program). As a result, it is responsible for setting the set-point for the desired position.
Local and remote set-point	Enables controlling the template with a set-point that is determined through a set-point selector (local or remote) in program owner. The local set-point is assigned from the sequential control while the remote set-point is assigned to the template from the logic implemented in the continuous/trigger based control.
Running status	Allows determining the running status of the element to be controlled and monitored with the help of run and stop feedback.
Interlocking	The template gives a command to the device to move to the defined position in case of an active interlock. Interlock bypass and setpoint reset options are available.
Mode selection	The object can be switched to different modes of operation like Normal, Override, Out of Service, Simulation and Maintenance.
Inoperable device	Allows monitoring inoperable device condition (optional). Template can be configured to withdraw output based on this condition and setpoint can be withdrawn in case of detected error.

Function	Description
Detection of unsuccessful operation	If position is unsuccessful within a configured time after the output command is activated, template detects unsuccessful operation.
Manual resetting	Allows you to reset manually after the detection of an inoperable device condition or unsuccessful operation. After reset, device output will follow set point. It also allows resetting all the rearm conditions of optional services as well on specific configuration.

Optional Services

Available optional services are:

- **Interlocks**
- **Failures**
- **Local Panel**
- **Maintenance**

Parameters

Parameters

The `$MotorGP` template provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default value	Additional remarks
Setpoint	**Reset on failure**	`Boolean`	*FALSE*	Resets the setpoint when the motor is non-operational.
	Track setpoint on interlock	`Boolean`	*TRUE*	The interlock setpoint is tracked by the equipment setpoint when the interlock condition is active.
	Force remote setpoint	`Boolean`	*FALSE*	Enables the remote control operation for motor.
Failure	**Confirmation time out**	`Dura-tion`	*00:00:03*	Maximum time for feedback confirmation.
	Minimum alarm duration		*00:00:05*	Minimum time to maintain the confirmation alarm or detected error alarm.
	Disable alarm when motor is not running	`Boolean`	*FALSE*	Disables the alarm, if motor stopped.
Rearm	**Required for failures**	`Boolean`	*TRUE*	Device requires a reset after device error or interlock trip alarm.
	Required on confirmation failure			Device requires a reset after confirmation error.
	Resets conditions		*FALSE*	Rearms the motor, interlock and inoperable conditions.

Inputs (Digital Input)

The table describes the digital input:

Name	Description
Running	It allows you to connect the run feedback of the motor.
Fail	It allows you to connect the inoperable condition of the motor.
Stopped	It allows you to connect the stop feedback of the motor.
ExternalControl	It allows you to control the device externally.

For more details, refer to the topic describing the parameters of digital inputs, page 44.

Outputs (Digital Output)

The table describes the digital output:

Name	Description
OP	It allows you to connect the motor output signal to the output channel.

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Failures

For more details, refer to the topic describing the parameters of failures, page 74.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Maintenance

For more details, refer to the topic describing the parameters of maintenance, page 89.

Local Panel (Logic)

For more details, refer to the topic describing the parameters of logic, page 45.

Local Panel (Digital Input)

The table describes the digital input:

Name	Description
LPMode	Local panel selection signal from mode selector switch.
ZeroMode	Zero/Off selection signal from mode selector switch.
CSMode	Control selection signal from selector switch.
OnPB	Motor start signal from local panel.
OffPB	Motor stop signal from local panel.

For more details, refer to the topic describing the parameters of digital inputs, page 44.

Local Panel (Digital output)

The table describes the digital output:

Name	Description
PBLight	Output signal to indicate that push buttons are operational.

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default value	Additional remarks
Operation	**Normal owner**	`Enum`	*Program and cascade (5)*	It allows defining the normal owner groups and any other owner who has logged in indicates any abnormality in graphics.
Security	**Set owner**	`Enum`	*Operator (1)*	Privilege to change owner. This parameter is used in Operator Tab, page 155.
	Set setpoint		*Operator (1)*	Privilege to change setpoint. This parameter is used in Operator Tab, page 155.
	Set mode - override		*Supervisor (2)*	Privilege to change override mode.This parameter is used in Engineering Tab, page 156.
	Set mode - simulation		*Engineer (5)*	Privilege to change simulation mode. This parameter is used in Engineering Tab, page 156.
	Set mode - maintenance		*Maintenance (3)*	Privilege to change maintenance mode. This parameter is used in Engineering Tab, page 156.
	Set mode - out of service		*Supervisor (2)*	Privilege to change out of service mode. This parameter is used in Engineering Tab, page 156.
	Set reset		*Operator (Confirmed) (10)*	Privilege to reset. This parameter is used in Operator Tab, page 155.
	Set alarm acknowledge		*Operator (1)*	Privilege to acknowledge alarm. This parameter is used in Alarm Tab, page 67.
	Set bypass conditions		*Supervision (2)*	Privilege to set bypass conditions mode. This parameter is used in Engineering Tab, page 156.
	Set bypass all interlocks		*Supervisor (2)*	Privilege to bypass all interlocks. This parameter is used in Engineering Tab, page 156.
	Set asset maintenance		*Operator (1)*	Privilege to set asset maintenance mode. This parameter is used in Operator Tab, page 155.
Events	**Enable log**	`Boolean`	*FALSE*	It allows to historize the change setpoints. Example: Running
Historize	**Owner/mode**	`Boolean`	*FALSE*	It allows to historize the change in mode and owner.

Asset and Display Parameters

For more details, refer to the topic describing the parameters of asset and display, page 43.

Alarm Parameters

For more details, refer to the topic describing the parameters of alarm, page 45.

Failures

For more details, refer to the topic describing the parameters of failures, page 74.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Maintenance

For more details, refer to the topic describing the parameters of maintenance, page 89.

Local Panel

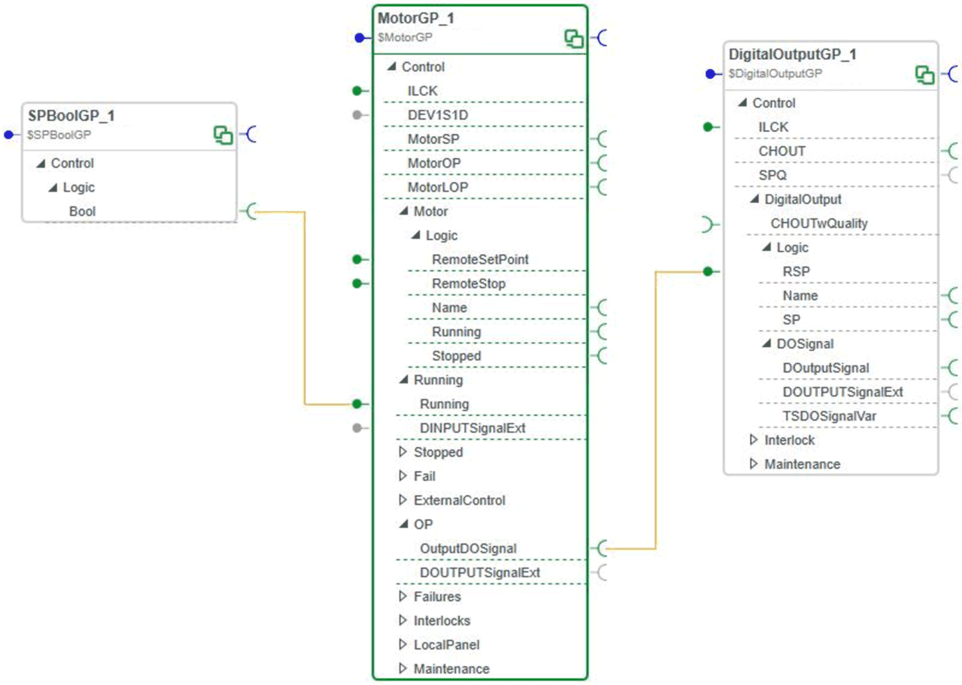
For more details, refer to the topic describing the parameters of local panel, page 44.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

The following figure depicts the $MotorGP (also applicable for $MotorCE) template as it appears in `Links Editor` and an example of other templates connected to it:



For interface identifiers, refer to the control module template interfaces, page 49.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

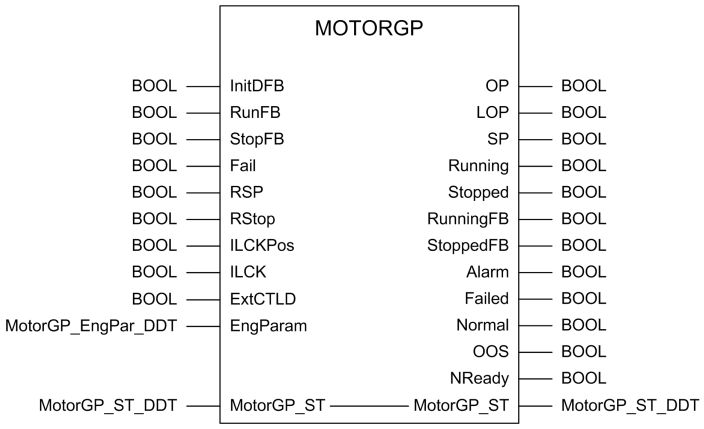
Overview

This section describes the `MOTORGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Name	Type	Description
`InitDFB`	`BOOL`	If a rising edge is detected on this input, the internal timers of the block are reset and detected alarms are set to 0.
`RunFB`	`BOOL`	1 = Indicates to the DFB that motor is running. Usually, it is used to connect the digital input of a motor for switching operation confirmation.
`StopFB`	`BOOL`	1 = Indicates to the DFB that motor is stopped. Usually, it is used to connect the digital input of a motor `Stop` signal.
`Fail`	`BOOL`	1 = Indicates to the DFB a detected error in the device. Usually used to connect the digital thermal trip input of a motor switch.
`RSP`	`BOOL`	Remote setpoint by continuous control (0 = Stop, 1 = Start). If `RStop` is connected, then the rising edge (Pulse based control) sends the command to start the motor only when program/remote cascade mode is configured.
`RStop`	`BOOL`	A rising edge sends a stop command only when program/remote cascade mode is configured.
`ILCKPos`	`BOOL`	1 = Position needs to be used when the DFB is interlocked. Refer to the `ILCK` input.
`ILCK`	`BOOL`	1 = Interlocks the device at the defined position. **NOTE:** If the motor requires resetting (`STW.ReamR` = 1), the `OP` takes the value = 0 regardless of the `ILCKPos` value waiting for the operator to carry out the reset.

Name	Type	Description
ExtCTLD	BOOL	1 = Indicates to the DFB if the device is being controlled from outside of the system. Monitoring of confirmation alarms is stopped during external control. The Fail input continues to be evaluated.
EngParam	MotorGP_EngPar_DDT	Engineering parameters

MotorGP_EngPar_DDT Type

Name	Type	Description
TimeOut	TIME	Enables configuring the maximum time that the device takes to reach the defined position. When the output is triggered, if the limit switch is not detected even after the TimeOut time is elapsed. The device is switched to detected alarm state if the limit switch is not detected. When it has a value of 0, monitoring is deactivated.
ScanTime	TIME	The minimum time for which the detected alarm signals are kept active. ScanTime confirms that the supervision system acquires the data for detected alarms that are automatically reset.
ConfRearmEn	BOOL	• 1 = Withdraws the output *OP* when a non-confirmed operation is detected (*Alarm* = 1) and requires a mandatory manual resetting once the *Alarm* condition disappears. • 0 = The output *OP* remains in its previous state when a non-confirmed operation is detected (*Alarm* = 1) and does not need manual resetting once the *Alarm* condition disappears. Refer the TimeOut parameter and the MotorGP_ST.STW.Alarm input/output.
FailRearmEn	BOOL	• 1 = A manual resetting is required when the detected error/interlock trip alarm condition disappears. • 0 = No manual resetting is required when the detected error/interlock trip alarm condition disappears.
DisStopA-larm	BOOL	1 = Disable device detected error alarm when motor is not running.
ResetSPFail	BOOL	Reset setpoint in case of detected error condition (1 = Restart is required if motor trips due to detected error or alarm).
TrackSPEn	BOOL	Track set-point in case of that the interlock condition is active. **NOTE:** The *TrackSPEn* parameter enables the interlock trip alarm when the device stopped due to interlock conditions.

Outputs

Output Parameter Description

Name	Type	Description
OP	BOOL	1 = Field Command signal active. Allows the control signal of the device to be connected. It always follows the LOP command. **NOTE:** During simulation the signal will be deactivated.
LOP	BOOL	1 = Logical command signal active. Refer to the LOP Calculation, page 145.

Name	Type	Description
SP	BOOL	1 = Current device set-point Refer to the SP Calculation, page 146.
Running	BOOL	1 = Determines if the motor has reached the position corresponding to the set-point (SP) with a value of 1. It is evaluated by considering any detectors. Refer to the output Stopped and input/output MotorGP_ST.CFGW.Sim, page 146. Refer the Running and Stopped feedback Calculation, page 146.
Stopped	BOOL	1 = Determines if the motor has reached the position corresponding to the set-point (SP) with a value of 0. It is evaluated by considering any detectors. Refer to the output Running and input/output MotorGP_ST.CFGW.Sim, page 146. Refer the Running and Stopped feedback Calculation, page 146.
RunningFB	BOOL	1 = Indicates whether the position corresponding to the set-point (SP) with a value of 1 has been reached. It is independent from the other detector if the StopFB detector is available. Refer the RunningFB and StoppedFB feedback Calculation, page 146.
StoppedFB	BOOL	1 = Indicates whether the position corresponding to the set-point (SP) with a value of 0 has been reached. It is independent from the other detector if the RunFB detector is available. Refer the RunningFB and StoppedFB feedback Calculation, page 146.
Alarm	BOOL	1 = The detection of unsuccessful operation has occurred. Refer the for monitoring details.
Failed	BOOL	1 = Indicates an inoperable device. It is evaluated according to the value of the detected error and if FailRearmEn engineering parameter configured then Failed will be high until Rearm is provided. When it is activated, it is timed so that the signal is maintained for a minimum of ScanTime. 0 = After the minimum activation time has passed, it is only deactivated if the detected error input signal is deactivated. **NOTE:** Detected error condition is considered even when the motor is in simulation mode.
Normal	BOOL	Indicates the motor is operating in normal mode.
OOS	BOOL	1 = Indicates the device is out of service.
NReady	BOOL	Indicates motor is not ready for operation.

LOP Calculation

The evaluation of the device control depending on the interlock status, the current set-point, and if the DFB requires resetting is shown in the following table:

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

The *ConfRearmEn* engineering parameter has to be set to 1 for the *OP* pin of the function block output to become 0, under non-confirmed operations.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: Non-confirmed operations are those for which commands have been given and to which there is no confirming reply from the equipment.

SP Calculation

The signal evaluation depending on the SC.Rem and RSP inputs and the MotorGP_ST.CFGW.Owner input/output is shown in the following table:

Owner	Rem	SP is calculated as
OFF	OFF	Sequential control (SC.LSP) or Pulse base control (SC.LStart, SC.LStop)
OFF	ON	Continuous control (RSP) or Pulse base control if RClose pin is connected (RSP, RStop)
ON	-	MotorGPST.CFGW.LSP

Feedback Calculation for Running and Stopped

RunFB connected	StopFB connected	Simulation	Override	Running is calculated as:	Stopped is calculated as:
-	-	-	ON	LOP	Not LOP
-	-	ON	OFF	LOP and (TimeOut/2)	Not LOP and (TimeOut/2)
OFF	OFF	OFF	OFF	LOP	Not LOP
OFF	ON	OFF	OFF	Not StopFB	StopFB
ON	OFF	OFF	OFF	RunFB	Not RunFB
ON	ON	OFF	OFF	RunFB and (not StopFB)	StopFB and (not RunFB)

Feedback Calculation for RunningFB and StoppedFB

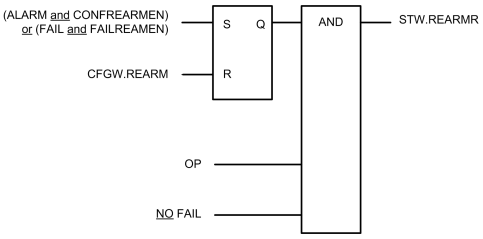
RunFB connected	StopFB connected	Simulation	Override	RunningFB is calculated as:	StoppedFB is calculated as:
-	-	-	ON	LOP	Not LOP
-	-	ON	OFF	LOP and (TimeOut/2)	Not LOP and (TimeOut/2)
OFF	OFF	OFF	OFF	LOP	Not LOP
OFF	ON	OFF	OFF	Not StopFB	StopFB
ON	OFF	OFF	OFF	RunFB	Not RunFB
ON	ON	OFF	OFF	RunFB	StopFB

Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
MotorGP_ST	MotorGP_ST_DDT	Data structure of the DFB status.

MotorGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-write access to the data contained in this word.
	RunningFB	*Bit0*	BOOL	Refer to RunningFB output pin, page 144.
	StoppedFB	*Bit1*	BOOL	Refer to StoppedFB output pin, page 144.
	SP	*Bit2*	BOOL	Refer to SP output pin, page 144.
	LOP	*Bit3*	BOOL	Refer to LOP output pin, page 144.
	Alarm	*Bit4*	BOOL	Refer to Alarm output pin, page 144.
	Failed	*Bit5*	BOOL	Refer to Failed output pin, page 144.
	Rem	*Bit6*	BOOL	• 1 = Enables the DFB to be configured to remote set-point Rem. • 0 = Enables the DFB to be configured to local set-point LSP.
	ILCK	*Bit7*	BOOL	Refer to ILCK input pin, page 143
	RearmR	*Bit8*	BOOL	Indicates if the device requires resetting (1) after the detection of an inoperable device condition if it has been configured through the ConfRearmEn and/or FailRearmEn inputs. The signal activation logic included in the DFB is as follows. 
STW	ExtCTLD	*Bit9*	BOOL	Read-only access Refer to ExtCTLD input pin, page 143
	NReady	*Bit10*	BOOL	Refer to NReady output pin, page 144.
	FailAlarm	*Bit11*	BOOL	1 = Detected error alarm is active.
	SPLocked	*Bit13*	BOOL	Indicates whether the setpoint change buttons in the monitoring (HMI) system should be locked (1) or not (0).
	ILCKTrip	*Bit15*	BOOL	1 = Interlock trip active
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read-only access to the data contained in this word.
	Owner	*Bit0*	BOOL	Read/write access Refer to Owner, page 148.
	ILCKBP	*Bit1*	BOOL	Read/write access 1 = Enables the interlock to be bypassed.
	Override	*Bit2*	BOOL	Refer to Override, page 148.
	Rearm	*Bit3*	BOOL	Write access Refer to Rearm, page 148.
	LSP	*Bit4*	BOOL	Read/write access Refer to LSP, page 148.
	SimFBFail	*Bit12*	BOOL	1 = Feedback will not simulate in simulation mode.

Nam-e	Bit Name	Bit	Type	Description
	Sim	*Bit13*	BOOL	Read/write access Refer to Sim, page 148.
	Maint	*Bit14*	BOOL	Refer to Maint, page 148.
	OOS	*Bit15*	BOOL	Read/write access Refer to OOS output pin, page 144.

Public Variables

Public Variable Description

Variable	Type	Description
SC	MotorGP_SC_DDT	Provides the frequently needed data to monitor the device status and to control it from the sequential control.

MotorGP_SC_DDT Type

Name	Type	Description
Rem	BOOL	Read/write access 1 = Enables the DFB to be configured to remote set-point RSP 0 = Enables the DFB to be configured to local set-point LSP.
LSP	BOOL	Read/write access 1 = Enables the assignment of the target local set-point usually for the sequential control if the owner is the Program (Owner input/output is OFF) and the selected set-point is Local (SC.Rem public variable is OFF). 0 = The block automatically assigns the current value of the resulting set-point. It is described as follows:

Owner	Rem	SC.LSP
OFF (Program)	OFF	SC.LSP (last value or new value assigned from the sequential control)
OFF	ON	SP (calculated from the RSP)
ON (Opera-tor)	ON	SP (last value or new value assigned by the Operator)

Name	Type	Description
LStart	BOOL	Read/write access 1 = Rising edge to enable Start command while the owner is the Program and Local.
LStop	BOOL	Read/write access 1 = Rising edge to enable Stop command while the owner is the Program and Local.
OP	BOOL	Read-only access Refer to OP output pin, page 144.
LOP	BOOL	Read-only access Refer to LOP output pin, page 144.
SP	BOOL	Read-only access Refer to SP output pin, page 144.

Name	Type	Description
Running	BOOL	Read-only access Refer to Running output pin, page 144.
Stopped	BOOL	Read-only access Refer to Stopped output pin, page 144.
RunningFB	BOOL	Read-only access Refer to RunningFB output pin, page 144.
StoppedFB	BOOL	Read-only access Refer to StoppedFB output pin, page 144.
Alarm	BOOL	Read-only access Refer to Alarm output pin, page 144.
Failed	BOOL	Read-only access Refer to Failed output pin, page 144.
FailAlarm	BOOL	Read-only access 1 = Detected error alarm is active (Read-only).
ILCKD	BOOL	Read-only access The signal evaluation depending on the ILCK input and the MotorGP_ST.CFGW.ILCKBP input/output is shown as follows: **ILCKD** / **ILCKBP** / **SC.ILCKD** OFF / OFF / OFF OFF / ON / OFF ON / OFF / ON ON / ON / OFF
Rearm	BOOL	Read/write access Allows the device to be reset (1). The block sets the signal to 0 after each execution and sets the signal indicating that the resetting is required (CFGW.REARM) to 0.
RearmR	BOOL	Read/write access Refer to RearmR, page 147.
Owner	REAL	Read/write access Refer to the MotorGP_ST.CFGW.Owner Input/output pin, page 146.
Normal	BOOL	Read/write access Refer to Normal output pin, page 144.
Override	BOOL	Read/write access Enables to Override mode.
Sim	BOOL	Read/write access Enables to simulation mode.
Maint	BOOL	Read-only access 1 = Maintenance mode active.
OOS	BOOL	Read/write access Refer to OOS output pin, page 144.
NReady	BOOL	Read-only access Refer to NReady output pin, page 144.
ExtCTLD	BOOL	Read-only access Refer to the ExtCTLD input pin, page 143.

Optional Service - Local Panel (MOTORLPGP)

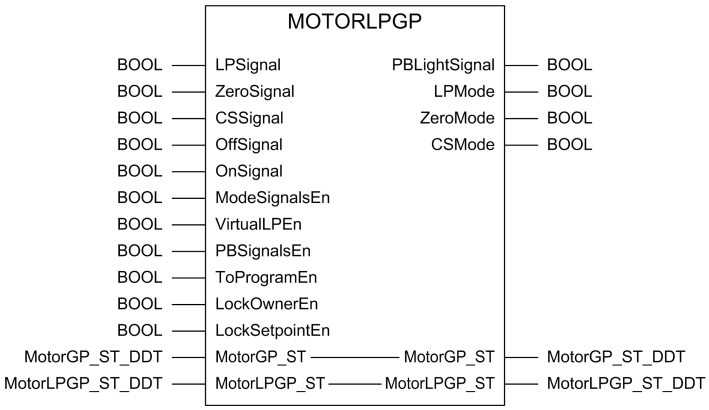
Overview

This chapter describes the optional services - local panel (MOTORLPGP) DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Name	Type	Description
`LPSignal`	`BOOL`	1 = The local panel mode switch is in the Local Panel position. Refer to the table included with the description of the `LPMode` output pin, page 151.
`ZeroSignal`	`BOOL`	1 = The local panel mode switch is in the zero position.This signal is optional even if a mode switch is available on the local panel. When this signal is a logical high, user is not allowed to operate the device from the faceplate and also from the DFB. Hence, all the functions in the faceplate will be disabled. Refer to the table included with the description of the `LPMode` output pin, page 151.
`CSSignal`	`BOOL`	1 = The local panel mode switch is in the Control System position. Refer to the table included with the description of the `LPMode` output pin, page 151.
`OffSignal`	`BOOL`	1 = The OFF push button on the local panel is pressed. **NOTE:** OFF push button signal has higher priority than ON push button signal.
`OnSignal`	`BOOL`	1 = The ON push button on the local panel is pressed.
`ModeSignalsEn`	`BOOL`	1 = Enables the use of the Local Panel/Zero/Control System (or Local Panel - Control System) mode switch on the local panel. Usually configured in development phase based on the characteristics of the local panel being used. If the mode switch is enabled, the `LPSignal`, `ZeroSignal`, and `CSSignal` inputs are considered to determine the operating modes of the local panel. Refer to the table included with the description of the `LPMode` output pin, page 151. **NOTE:** *ModeSignalsEn* input pin has higher priority than the *VirtualLPEn* input pin signal.
`VirtualLPEn`	`BOOL`	1 = This signal enables the operator to select the Local/Control System mode of operation from the HMI, so that the push button signals are enabled for operation. **NOTE:** When the owner is program, the Local/Control System mode selection drop-down list will be visible but disabled for operation however, when the owner is operator the Local Panel drop-down list will be accessible for operation. Refer to the table included with the description of the `LPMode` output pin, page 151.

Name	Type	Description
PBSignalsEn	BOOL	1 = Enables to accept the OFF and ON push button signals from the local panel. This signal is not applicable if the mode switch is enabled (refer to the ModeSignalsEn input). It means that push button signals are considered when in Local Panel mode. When the mode switch is disabled (ModeSignalsEn = 0), the PBSignalsEn signal enables/disables the push button signals from the control system itself based on the relevant process conditions. The SC.DisableLP public variable is also considered for determining whether the push buttons are enabled or not when *LPMode* pin signal is high, as shown in the following table.

VirtualLPEn	ModeSignalsEn	PBSignalsEn	SC.DisableLP	Push buttons enabled?
OFF	OFF	OFF	-	NO
OFF	OFF	-	ON	NO
OFF	OFF	ON	OFF	YES
OFF	ON	-	-	YES
ON	OFF	OFF	-	YES

Therefore, this signal can be used to enable/disable the push buttons on a simple local panel that does not feature a mode switch (Local Panel/Zero/Control System) based on relevant process conditions.

Name	Type	Description
ToProgramEn	BOOL	1 = Change to program mode when control is switched to control system mode and VirtualLPEn = 0.
LockOwnerEn	BOOL	1 = Disables the access of changing the owner from HMI while the mode switch is in local panel. The following table shows how the on-off device owner is evaluated based on *ToProgramEn*, *LockOwnerEn* input signals and the local panel operating mode:

Local Panel Mode	ToProgramEn	LockOwnerEn	Owner
Zero mode	-	-	Operator
Switch to Local Panel	-	-	Operator
Local Panel	-	ON	Operator
Switch to Control System	ON	-	Program

In remaining cases, owner of the on/off device ceases to be defined from the MotorLPGP DFB and it enables to change it from monitoring (HMI) system with the command word in MotorLPGP_ST.CFGW input/output of the corresponding MotorLPGP DFB.

In any case, consider that, the device owner that had been set before switching to the local panel mode is not memorized.

Name	Type	Description
LockSetpointEn	BOOL	1 = Disables the access to set point of on/off devices in the monitoring (HMI) system when it is in local panel mode only, and the owner is operator. Therefore, user can activate this signal to lock setpoint operation from the monitoring (HMI) system while on/off device is controlled from the local panel.

Outputs

Output Parameter Description

Name	Type	Description
PBLightSignal	BOOL	1 = The push buttons on the local panel are fully operational, that is, the local panel is in Local Panel mode (either because the mode is selected with the corresponding switch or because the push buttons are enabled in the event that there is no selector switch available or the *VirtualLPEn* is high with the local panel mode enabled).
LPMode	BOOL	1 = The local panel mode is in Local Panel. The following table shows how the local panel operating mode is determined based on the ModeSignalsEn, VirtualLPEn, PBSignalsEn, LPSignal, CSSignal and ZeroSignal, input signals:

ModeSignalsEn	VirtualLPEn	PBSignalsEn	LPSignal	CSSignal	ZeroSignal	OUTPUT
0	0	0	0	0	0	CS mode
1	0	0	0	0	0	Zero mode

Name	Type	Description						
		1	0	0	1	0	0	LP mode and PBLightSignal
		1	0	0	0	1	0	CS mode
		0	1	1	NA	NA	NA	CS mode
		0	1	0	NA	NA	NA	CS mode
		0	0	1	0	0	0	LP mode and PBLightSignal
		1	1	0	1	0	0	LP mode and PBLightSignal
		1	1	0	0	0	0	Zero mode
		1	1	0	0	0	1	Zero mode
		1	1	0	0	1	0	CS mode
		1	0	1	0	1	0	CS mode
		1	1	1	0	1	0	CS mode
		1	0	0	0	0	1	Zero mode
ZeroMode	BOOL	1 = The local panel mode is in Zero. User is not allowed to operate the device from the faceplate and also from DFB. Hence, all the functions in the faceplate will be disabled. Refer to the table included with the description of the LPMode output.						
CSMode	BOOL	1 = The local panel mode is in Control System. Refer to the table included with the description of the LPMode output.						

Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
MotorGP_ST	MotorGP_ST_DDT, page 146	Data structure corresponding to the motor is to be controlled from the local panel (for detailed information regarding this structure, refer to the MOTORGP DFB, page 143. The structure provides the information needed for the operation of the MOTORLPGP DFB and enables this DFB to control the owner and set-point for the on-off device.
MotorLPGP_ST	MotorLPGP_ST_DDT	Data structure is used as an interface with the monitoring (HMI) system.

MotorLPGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	LPMode	*Bit0*	BOOL	Local panel in Local Panel mode. Refer to the LPMode output pin, page 151.
	ZeroMode	*Bit1*	BOOL	Local panel in zero mode. Refer to the ZeroMode output pin, page 151.
	CSMode	*Bit2*	BOOL	Local panel in Control System mode. Refer to the CSMode output pin, page 151.
	OwnerLocked	*Bit3*	BOOL	Indicates whether the owner change buttons in monitoring (HMI) system needs to be locked (1) or not (0). Is activated when: • The Local Panel is set to zero mode. • The local panel is set to local panel mode and the *LockOwnerEn* input signal has been configured as active (1).

Name	Bit Name	Bit	Type	Description
	SPLocked	*Bit4*	BOOL	Indicates whether the set-point change buttons in the monitoring (HMI) system should be locked (1) or not (0). Is activated when: • The Local Panel is set to zero mode. • The local panel is set to local panel mode and the *LockSetpointEn* input signal has been configured as active (1).
	ModeSignalsEn	*Bit5*	BOOL	Indicates the state of the ModeSignalsEn input signal, that is, whether there is a mode switch (1) or not (0).
	PBEnabled	*Bit6*	BOOL	Indicates whether the push buttons of the local panel are enabled (1) or not (0). The signal is activated when at least one of the ModeSignalsEn and PBSignalsEn signal inputs is active.
	VirtualLPEn	*Bit7*	BOOL	Enables the drop-down list for the Local Panel / Control System mode selection.
	LPSignal	*Bit8*	BOOL	Indicates the state of the LPSignal input signal.
	ZeroSignal	*Bit9*	BOOL	Indicates the state of the ZeroSignal input signal.
	CSSignal	*Bit10*	BOOL	Indicates the state of the CSSignal input signal.
	OffSignal	*Bit11*	BOOL	Indicates the state of the OffSignal input signal.
	OnSignal	*Bit12*	BOOL	Indicates the state of the OnSignal input signal.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read-only access to the data contained in this word.
	LPEnabled	*Bit0*	BOOL	• 1 = Enables the Local Panel mode. • 0 = Enables the Control System mode.

Public Variables

Public Variable Description

Variable	Type	Description
SC	MotorLPGP_SC_DDT	Provides the frequently needed data for monitoring and controlling the DFB.

MotorLPGP_SC_DDT Type

Name	Type	Description
LPMode	BOOL	Read-only access Refer to the LPMode output pin, page 151.
ZeroMode	BOOL	Read-only access Refer to the ZeroMode output pin, page 151.
CSMode	BOOL	Read-only access Refer to the CSMode output pin, page 151.
DisableLP	BOOL	Read/write access 1 = Disables the push buttons on the local panel. Only applicable when a local panel without mode switch configuration is used (ModeSignalsEn = 0).

Supervision

Overview

This section describes the Supervision functions and runtime services that are available for the motor.

Genies

Genie Properties

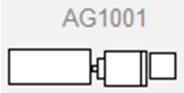
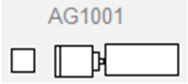
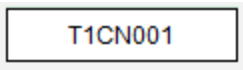
Refer to Genie Properties, page 62

Genie Icons

Refer to Genie Icons, page 63

Representation

The table shows the genies of the `gpl_motor` library:

Graphic symbol	Genie name	Graphic symbol	Genie name
	Blower_Left		Blower_Right
	Motor		Pump_Left
	Pump_Right		Rotary_Valve
	Screwpump_Left		Screwpump_Right
	Motor_Up		motorinstanceid_m

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a general description of the graphic elements that are used in the faceplate.

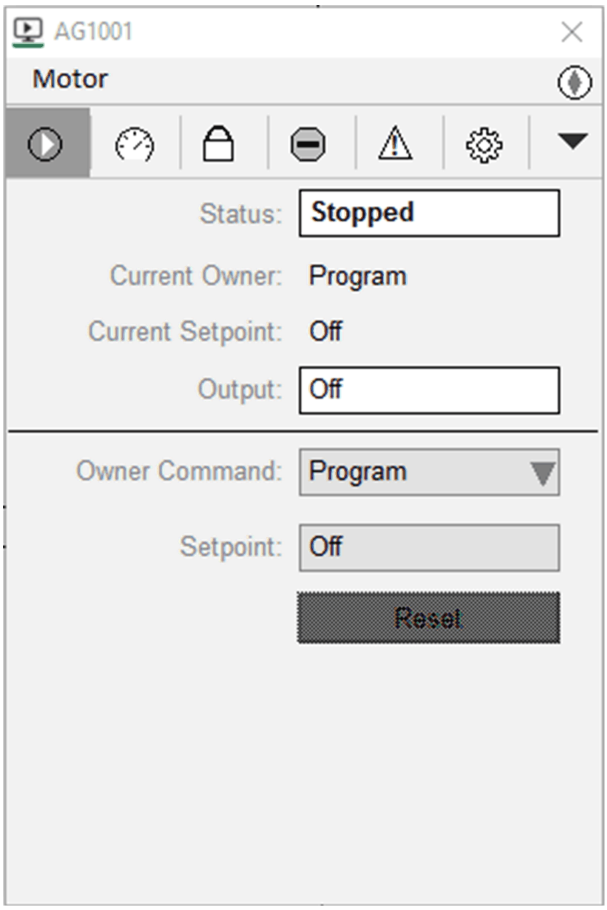
Available Tabs

During operation, clicking an motor genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
  - Event, page 69
- Tabs for optional functions, which appear only if configured:
  - Interlock, page 87
  - Failure, page 78
  - Measures, page 70

Operator Tab

The figure shows an operator tab:

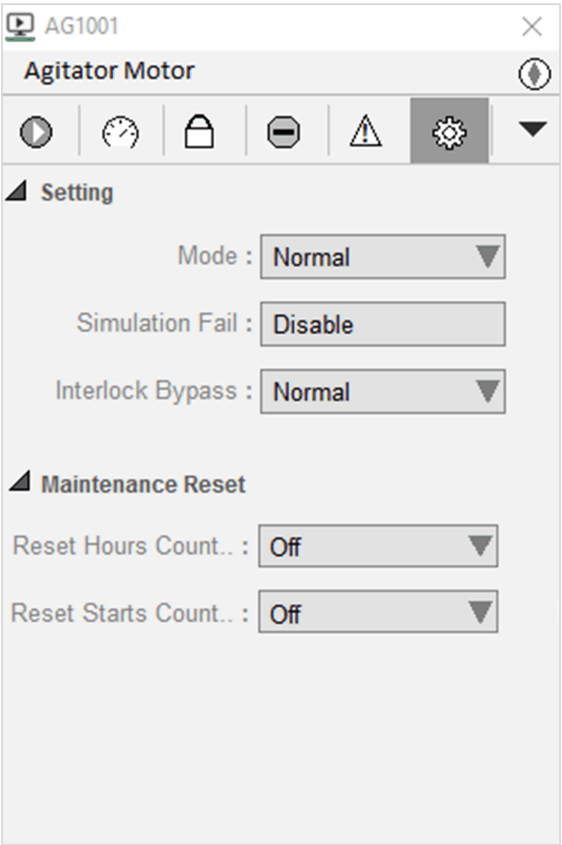


The table describes the control function of an operator tab:

Label	Item name	Description	Security parameter, page 141
Status	EquipStatus	Equipment status of motor. For more details, refer to EquipStatus calculated variable, page 159.	—
Current Owner	Owner	Current owner of motor. For more details, refer to operating owner, page 58.	
Current Setpoint	Setpoint	Current setpoint of motor.	
Output	Equipstate	Output status of motor. For more details, refer to EquipState calculated variable, page 159.	
Owner Command	OwnerCommand	Owner command of motor (Program, Operator or Local Panel). **NOTE:** Local panel owner is enabled when the virtual panel is selected.	Set owner
Setpoint	CommandSet-point	Allows you to change the setpoint, if owner = operator.	Set setpoint
–	Rearm	Allows you to reset manually after the detection of an inoperable device condition or unsuccessful operation. **NOTE:** When the control module is reset, the current setpoint shown in operator tab is effective. Also, when you click **Reset**, a dialog box opens to confirm.	Set rearm

Engineering Tab

The figure shows an engineering tab:



The table describes the control function of engineering tab:

Label	Item name	Description	Security parameter, page 141
Setting			
Mode	Mode	Allows you to view and change the mode. For more details, refer to operating mode, page 59.	Set mode - normal, set mode - override, set mode - simulation, set mode - maintenance, and Set mode - out of service **NOTE:** Each configurable mode has the different security parameters
Simulation Fail	SimulationFailActive	Enable or disable feedback simulation. **NOTE:** The Simulation Fail will only be available when the parameter `SimulationFail` is created in the citect parameters. To create, go to **Setup** > **Parameters**. Enter the `GPL.Mode` in **Section Name**, `SimulationFail` in **Name** and set the **Value** = 1.	Set mode - simulation
Interlock Bypass	InterlockBypass	Interlock bypass status Allows you to bypass interlock.	Set bypass conditions, set bypass all interlocks
Maintenance Reset			
Reset Hours Count	MotMntResetHour-sCounter	Reset the hours counter.	Set reset
Reset Starts Count	MotMntResetManeuver-sCounter	Reset the operations counter.	Set reset

Items

Overview

This section describes the variables, equipstatus and equipstate variable, abnormal and action required conditions, alarms, events, and trends items of `motor`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`FeedbackRunning`	Run feedback	Running/Not Running	*Bit0*
`FeedbackStopped`	Stop feedback	Running/Not Running	*Bit1*
`Setpoint`	Selected setpoint	On/Off	*Bit2*
`Output`	Motor output status		*Bit3*
`AlarmActive`	Feedback confirmation detected error	Active/Inactive	*Bit4*
`FailActive`	Motor non-operational	Active/Inactive	*Bit5*
`ProgramSetpointMode`	Setpoint source for program owner	Remote/Sequence	*Bit6*
`InterlockActive`	Interlock status	Interlocked/Healthy	*Bit7*
`RequireRearm`	Reset required	Required/Not Required	*Bit8*
`OwnerExternal`	External owner status	External/Not External	*Bit9*

Item name	Description	Enumeration	Address
NotReady	Not ready to function	–	*Bit10*
FailAlarmActive	Detected error Alarm is active		*Bit11*
SetpointLockActive	Setpoint change is denied		*Bit13*
InterlockTrip	Interlocked trip is active		*Bit15*
Bit items derived from ConfigurationWord.			
OwnerSelect	Owner of control module	Operator/Program	*Bit0*
InterlockBypass	Interlock bypass command	Bypassed/Normal	*Bit1*
InputSource	Input source selection	Operator/Field	*Bit2*
Rearm	Rearm motor	Active/Inactive	*Bit3*
CommandSetpoint	Operator setpoint	On/Off	*Bit4*
SimulationFailActive	Feedback detected error in simulation mode	Enable/Disable	*Bit12*
Simulation	Simulation mode	–	*Bit13*
Maintenance	Maintenance mode		*Bit14*
OutOfService	Motor unavailable	Out Of Service/In Service	*Bit15*
IO Items			
StatusWord	Status word	–	MotorGP_ST.STW
ConfigurationWord	Configuration word		MotorGP_ST.CFGW
Calculated Items			
EquipStatus	Current status of the motor For more details, refer to EquipStatus calculated variable, page 159.	–	GPL_MotorStatus
EquipState	Output information For more details, refer to EquipState calculated variable, page 159.		GPL_MotorEquipState
Owner	Current owner of the motor For more details, refer to operating owner, page 58.		GPL_ OwnerExternalLocal
Mode	Current mode of the motor For more details, refer to mode function, page 59.		GPL_Mode
Abnormal	Shows abnormal condition of the motor For more details, refer to abnormal conditions, page 159.		GPL_Abnormal
ActionRequired	Shows abnormal condition of the motor For more details, refer to action required conditions, page 160.		GPL_ActionRequired
Optional Items			
Local panel			
Bit items derived from LPStatusWord.			
LocalPanelMode	Control from field	Enabled/Disabled	*Bit0*
ZeroMode	Motor control disabled		*Bit1*
ControlSystemMode	Control from remote		*Bit2*
LockedOwner	Owner change disabled from faceplate	Locked/Unlocked	*Bit3*
LockedSetpoint	Setpoint change disabled from faceplate		*Bit4*
LocalPanelEnabled	Control from field	Enabled/Disabled	*Bit5*
PushButtonEnabled	Local panel push buttons		*Bit6*

Item name	Description	Enumeration	Address
VirtualLocalPanelEnabled	Local panel enable from control		*Bit7*
LocalPanelSelected	Control from field	Selected/Not Selected	*Bit8*
ZeroSelected	Control disabled		*Bit9*
ControlSystemSelected	Control from remote		*Bit10*
PushButtonStop	Stop push button signal	On/Off	*Bit11*
PushButtonStart	Start push button signal		*Bit12*
Bit items derived from LPConfigurationWord.			
VirtualSwitch	Local panel selection from faceplate	Local/Control System	*Bit0*
IO Items			
LPStatusWord	Local panel status word	–	MOTORLPGP_ST.STW
LPConfigurationWord	Local panel configuration word		MOTORLPGP_ST.CFGW
Failures	Refer to the topic describing the variable items of failures, page 61.		
Interlocks	Refer to the topic describing the variable items of interlocks, page 60.		
Maintenance	Refer to the topic describing the variable items of maintenance, page 59.		

EquipStatus Calculated Variable

The table describes the *EquipStatus* calculated variable:

Equipment Status	Conditions	Enumeration
–	Run x Stop	Unknown
Motor Stopped	Not OP x Stop	Stopped
Motor Transition	Run x OP	Running
	OP x Not Run	Starting
	Not OP x Not Stop	Stopping

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	Conditions	Enumeration
Passive with Out of Service	OOS	Off
Passive to Active	Not OOS x Maint x OP	On
Passive with Out of Service	Not OOS x Maint x Not OP	Off
Passive	Not OOS x Not Maint x Fail	Fail
Active (Start) with Interlock	Not OOS x Not Maint x Not Fail x ILCK x OP	On Interlocked
Passive with Interlock	Not OOS x Not Maint x Not Fail x ILCK x Not OP	Interlocked
Passive to Active	Not OOS x Not Maint x Not Fail x Not ILCK x OP x Run	On
Active	Not OOS x Not Maint x Not Fail x Not ILCK x OP x Not Run	On
Active to Passive	Not OOS x Not Maint x Not Fail x Not ILCK x Not OP x Stop	Off
Passive	Not OOS x Not Maint x Not Fail x Not ILCK x Not OP x Not Stop	Off

Abnormal Conditions

The table describes the abnormal conditions of motor:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
`AbnormalOwner`	Abnormal owner status appears if the current owner is not a normal owner which is configured at the instance level.	*Bit0*
`AbnormalMode`	Abnormal mode status appears if the current operating mode is not normal mode.	*Bit1*
`AbnormalInterlockBy-pass`	Abnormal interlock bypass condition appears if the Interlock bypass is enabled.	*Bit2*
`AbnormalOperatorTab`	Abnormal status of operator tab. This condition appears if the Interlock bypass is enabled.	*Bit8*
`AbnormalFailTab`	Abnormal status of error tab.	*Bit10*
`AbnormalInterlockTab`	Abnormal status of interlock tab.	*Bit11*
`AbnormalEngineeringTab`	Abnormal status of engineering tab.	*Bit12*

Action Required Conditions

The table describes the action required conditions of motor:

Item name	Description	Address
Bit items derived from ActionRequired variable word.		
`ActionOperatorTab`	Action required status of operator tab.	*Bit8*
`ActionFailTab`	Action required status of error tab.	*Bit10*
`ActionInterlockTab`	Action required status of interlock tab.	*Bit11*

Alarms

The table describes the advanced alarm items that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
`MotMntOperationHour-sAlm`	Running hours counter overflow	`GPL_BitValueCheck (StatusWord, 0)`
`MotMntManeuversNum-bersAlm`	Operations counter overflow	`GPL_BitValueCheck (StatusWord, 1)`
`Alarm`	Detection of unsuccessful operation	`GPL_BitValueCheck (StatusWord, 4)`
`Fail`	Inoperable motor	`GPL_BitValueCheck (StatusWord, 11)`
`InterlockTrip`	Interlock trip	`GPL_BitValueCheck (StatusWord, 15)`

Events

Events are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
`StateAlarm`	All changes in the motor status and output.
`ModeAlarm`	All changes in the mode of operation of the equipment.
`ActionAlarm`	All the operator actions in the supervision.

Trends

No trend item is used by Supervision components for `motor`.

Dual Speed and Dual Direction Motor

What's in This Chapter

Template 162
Template - $Motor2SpdGP 167
Control 172
Optional Service - Local Panel (MOTOR2LPGP) 182
Supervision 186

Overview

This chapter describes the components that provide the functionality of template, control services, and Supervision functions of the dual speed and dual direction motor.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$Motor2DirGP` template.

Description

General Description

The `$Motor2DirGP` process control module template is used to control dual direction motor.

The process template allows the managing associated motors from the sequential control, the continuous control, and/or the supervision system depending on their configuration and the system needs.

Function Description

The main functions of the `$Motor2DirGP` template are described in the following table:

Function	Description
Owner selection	The object manages control system level, which is the owner (operator or program). As a result, it is responsible for setting the set-point for the desired position.
Local and remote setpoint	Enables the template with a set-point (stop/run and speed/rotation direction) that is determined through a set-point selector (local or remote) in program owner. The local set-point is assigned from the sequential control while the remote set-point is assigned to the template from the logic implemented in the continuous control.
Running status	Allows to determine the running status of the element to be controlled and monitored with the help of two limit switch (`Run1FB` and `Run2FB`).
Inoperable motor	Allows to monitor inoperable device condition. Template can be configured to withdraw output based on this condition and setpoint can be withdrawn in case of detected error.
Interlocking	The template gives a command to the device to move to the defined position in case of an active interlock. An interlock bypass and Setpoint is reset in case of interlock function is available.

Function	Description
Setpoint blocking	You can configure the template to avoid the activation of a certain setpoint (rotation direction or speed) depending on the conditions (for example, to avoid reverse speeds from being activated when the motor is already rotating in the forward direction).
Mode selection	The object can be switched to different modes of operation like Simulation, Maintenance, Out Of Service and Override.
Detection of unsuccessful operation	If position is unsuccessful within a configured time after the output command is activated, template detects unsuccessful operation.
Manual resetting	Allows you to reset manually after the detection of an inoperable device condition or unsuccessful operation. After reset, device output will follow set point. It also allows to reset all the rearm conditions of optional services as well on specific configuration.

Optional Services

Available optional services are:

- **Failures**
- **Interlocks**
- **Local Panel**
- **Maintenance**

Parameters

Parameters

The `$Motor2DirGP` template provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default value	Additional remarks
Setpoint	**Reset on Failure**	`Boolean`	*FALSE*	Resets the setpoint when the motor is inoperable.
	Track setpoint on interlock	`Boolean`	*TRUE*	The interlock setpoint is tracked by the equipment setpoint when the interlock condition is active.
	Force Remote Setpoint	`Boolean`	*FALSE*	Enables the remote control operation for motor.
Failure	**Confirmation Timeout**	`Duration`	*00:00:03*	Maximum time for feedback confirmation.
	Minimum Alarm Duration		*00:00:05*	Minimum time to maintain the confirmation alarm or detected error alarm.
	Disable Alarm when Motor is Not Running	`Boolean`	*FALSE*	Disables the alarm if motor is stopped.
Rearm	**Required for failure**	`Boolean`	*TRUE*	Device requires a reset after device error or interlock trip alarm.
	Required for Confirmation Failure			Device requires a reset after confirmation error.

Section	Parameter description	Type	Default value	Additional remarks
	Rearm Resets Conditions		*FALSE*	Rearms the motor, interlock and detected error conditions.

Inputs (Digital Input)

The table describes the digital input:

Name	Description
`ForwardRunning`	It allows you to connect the forward direction feedback of the motor.
`ReverseRunning`	It allows you to connect the reverse direction feedback of the motor.
`Fail`	It allows you to connect the detected error condition of the motor.
`ExternalControl`	It allows you to control the device externally.

For more details, refer to the topic describing the parameters of digital inputs, page 44.

Outputs (Digital Output)

The table describes the digital output:

Name	Description
`OP1`	It allows you to connect the motor output 1 signal to the output channel.
`OP2`	It allows you to connect the motor output 2 signal to the output channel.

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Failures

For more details, refer to the topic describing the parameters of failures, page 74.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Maintenance

For more details, refer to the topic describing the parameters of maintenance, page 89.

Local Panel (Logic)

For more details, refer to the topic describing the parameters of logic, page 45.

Local Panel (Digital Input)

The table describes the digital input:

Name	Description
`LPMode`	Local panel selection signal from mode selector switch.
`ZeroMode`	Zero/Off selection signal from mode selector switch.
`CSMode`	Control selection signal from selector switch.
`OFF`	Motor stop signal from local panel.
`ForwardS1`	Forward push button from local panel.
`ReverseS2`	Reverse push button from local panel.

For more details, refer to the topic describing the parameters of digital inputs, page 44.

Local Panel (Digital output)

The table describes the digital output:

Name	Description
PBLight	Output signal to indicate that push buttons are operational.

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default value	Additional remarks
Operation	**Normal Owner**	Enum	*Program and Cascade (5)*	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	**Set Owner**	Enum	*Operator (1)*	Privilege to change owner. This parameter is used in Operator Tab, page 220.
	Set Setpoint		*Operator (1)*	Privilege to change setpoint. This parameter is used in Operator Tab, page 220.
	Set Mode - Override		*Supervision (2)*	Privilege to change override mode.This parameter is used in Engineering Tab, page 221.
	Set Mode - Simulation		*Engineer (5)*	Privilege to change simulation mode. This parameter is used in Engineering Tab, page 221.
	Set Mode - Maintenance		*Maintenance (3)*	Privilege to change maintenance mode. This parameter is used in Engineering Tab, page 221.
	Set Mode - Out of Service		*Supervisor (2)*	Privilege to change out of service mode. This parameter is used in Engineering Tab, page 221.
	Set Reset		*Operator (Confirmed) (10)*	Privilege to reset. This parameter is used in Operator Tab, page 220.
	Set Alarm Acknowledge		*Operator (1)*	Privilege to acknowledge alarm. This parameter is used in Alarm Tab, page 67.
	Set Bypass Conditions		*Supervisor (2)*	Privilege to set bypass conditions mode. This parameter is used in Engineering Tab, page 221.
	Set Bypass All Interlocks		*Supervisor (2)*	Privilege to bypass all interlocks. This parameter is used in Engineering Tab, page 221.
	Set Asset Maintenance		*Operator (1)*	Privilege to set asset maintenance mode. This parameter is used in Engineering Tab, page 221.
Events	**Enable log**	Boo-lean	*FALSE*	Enable event logging for asset status.
Historize	**Setpoints**	Boo-lean	*FALSE*	It allows to historize the change setpoints. Example: Running

Section	Parameter description	Type	Default value	Additional remarks
	Owner/Mode			It allows to historize the change in mode and owner.
Input Signals				
Configu-ration	**Disable Energy**	Boo-lean	*TRUE*	–
	Energy engineering unit	String	*kWh*	–
	Disable load	Boo-lean	*TRUE*	–
	Load engineering unit	String	*A*	–
Trend	**Disable engineering**	Boo-lean	*TRUE*	–
	Disable load	Boo-lean	*TRUE*	–
Historize	**Energy**	Boo-lean	*FALSE*	It allows to historize the change in Energy.
	Load	Boo-lean	*FALSE*	It allows to historize the change in load.

Asset and Display Parameters

For more details, refer to the topic describing the parameters of asset and display, page 43.

Alarm Parameters

For more details, refer to the topic describing the parameters of alarm, page 45.

Failures

For more details, refer to the topic describing the parameters of failures, page 74.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Maintenance

For more details, refer to the topic describing the parameters of maintenance, page 89.

Local Panel

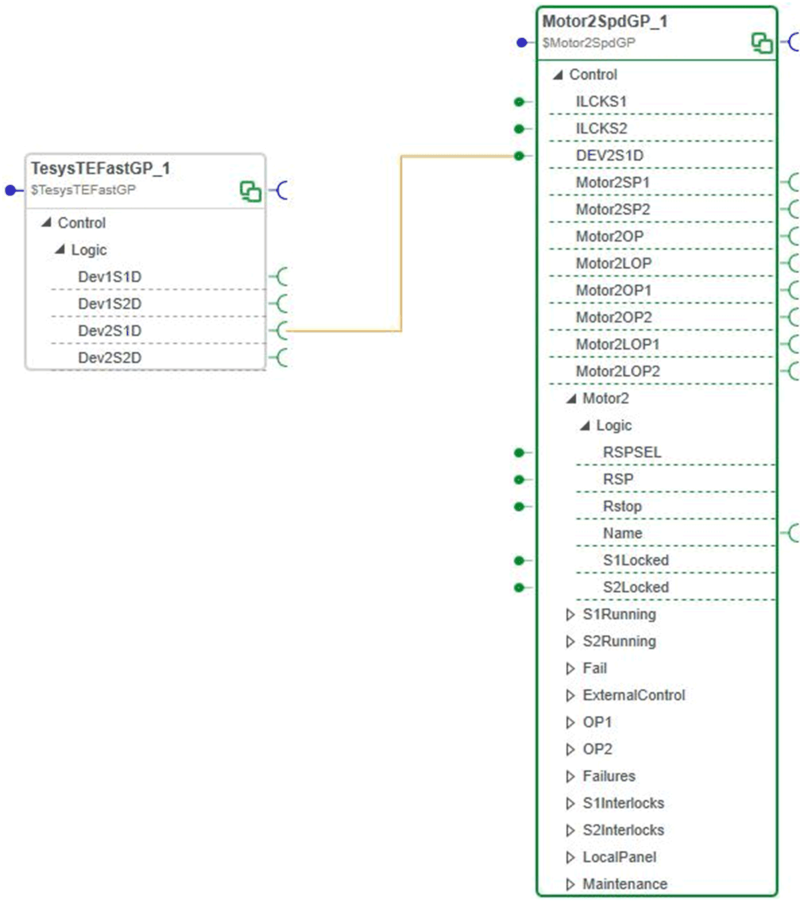
For more details, refer to the topic describing the parameters of local panel, page 44.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

The following figure depicts the `$Motor2DirGP$Motor2DirMMM` (also applicable for `$Motor2DirCE`)template as it appears in `Links Editor` and an example of other templates connected to it:



For interface identifiers, refer to the control module template interfaces, page 49.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Template - $Motor2SpdGP

Overview

This section describes the functionality, parameters, and interfaces of the `$Motor2SpdGP` template.

Description

General Description

The `$Motor2spdGP` process control module template is used to control on/off motors with two speeds (for example, slow speed and fast speed).

The process template allows the managing associated motors from the sequential control, the continuous control, and/or the Supervision system depending on their configuration and the system needs.

Function Description

The main functions of the template are described in the following table:

Function	Description
Owner selection	The template manages control system level, which is the owner (operator or program). As a result, it is responsible for setting the set-point for the desired position.
Local and remote setpoint	Enables the template with a set-point (stop/run and speed/rotation direction) that is determined through a set-point selector (local or remote) in program owner. The local set-point is assigned from the sequential control while the remote set-point is assigned to the template control from the logic implemented in the continuous control.
Running status	Allows to determine the actual position of the element to be controlled and monitored with the help of two feedback (`Run1FB` and `Run2FB`).
Interlocking	The template gives a command to the device to move to the defined position in case of an active interlock. An interlock bypass function is available.
Setpoint blocking	You can configure the template to avoid the activation of a certain setpoint (rotation direction or speed) depending on the conditions evaluated outside the template (for example, to avoid reverse speeds from being activated when the motor is already rotating in the forward direction).
Simulation	You can switch the template to Simulation mode. Thus, the operation confirmations for the controller motor are considered to be the same as the desired confirmations regardless of the actual signals for detection of not confirmed operation and inoperable device.
Inoperable motor	Allows to monitor inoperable device condition. Template can be configured to withdraw output based on this condition and setpoint can be withdrawn in case of detected failure.
Detection of not confirmed operation	If position is not confirmed within a configured time after the output command is activated, template detects not confirmed operation.
Manual resetting	Allows you to reset manually after the detection of an inoperable device condition or a not confirmed operation. After reset, device output will follow set point.

Optional Services

Available optional services are:

- Failure
- Interlocks
- Local Panel
- Maintenance

Parameters

Parameters

The `$Motor2spdGP` template provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Setpoint	Reset on Failure	`Boolean`	False	Resets the current setpoint when the motor is inoperable.

Section	Parameter description	Type	Default	Additional remarks
	Track setpoint on interlock			Track set-point in case of interlock condition is active.
	Force Remote Setpoint			An option to control the motor always remotely with any configuration.
Failure	Confirmation Timeout	`Duration`	00:00:03	Maximum time for feedback confirmation.
	Minimum Alarm Duration		00:00:05	Minimum time to maintain the confirmation alarm or detected fail alarm.
	Disable Alarm when Motor is Not Running	`Boolean`	False	It disable the alarms in case of non operational motor.
Rearm	Required for Failure	`Boolean`	True	A device becomes inoperable when a detected failure condition becomes high. A detected failure condition can be a single digital input, a signal from a drive or a summary of signals (defined as failures). Once a detected failure has occurred the motor can be prevented from starting until the operator rearms it from the operator tab.
	Required for Confirmation Failure			A confirmation detected failure occurs when the motor is unable to start or stop within the expected time (confirmation timeout). Once a confirmation detected failure has occurred the motor can be prevented from starting until the operator rearms the asset from the operator tab.
	Rearm Resets Conditions		False	Rearms the motor, interlock and detected failure conditions.

Inputs (Digital Input)

The table describes the digital input:

Name	Description
`Speed1RunningSignal`	It allows you to connect the fast speed1 feedback of the motor.
`Speed2RunningSignal`	It allows you to connect the low speed2 feedback of the motor.
`Fail`	It allows you to connect the detected fail condition of the motor.
`ExternalControl`	It allows you to control the device externally.

For more details, refer to the topic describing the parameters of digital inputs, page 44.

Outputs (Digital Output)

The table describes the digital output:

Name	Description
`OP1`	It allows you to connect the motor output signal to the output 1 channel.
`OP2`	It allows you to connect the motor output signal to the output 2 channel.

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Failures

For more details, refer to the topic describing the parameters of failures, page 74.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Maintenance

For more details, refer to the topic describing the parameters of maintenance, page 89.

Local Panel (Logic)

For more details, refer to the topic describing the parameters of logic, page 45.

Local Panel (Digital Input)

The table describes the digital input:

Name	Description
LPMode	Local panel selection signal from mode selector switch.
ZeroMode	Zero/Off selection signal from mode selector switch.
CSMode	Control selection signal from selector switch.
OffPB	Motor start signal from local panel.
ForwardS1	Speed 1 push button from local panel.
ReverseS2	Speed 2 push button from local panel.

For more details, refer to the topic describing the parameters of digital inputs, page 44.

Local Panel (Digital output)

The table describes the digital output:

Name	Description
PBLight	Output signal to indicate that push buttons are operational.

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Operation	Normal Owner	Enum	Program and Cascade (5)	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	Set Owner	Enum	Operator (1)	Privilege to change owner. This parameter is used in Operator Tab, page 188.
	Set Setpoint		Operator (1)	Privilege to change setpoint. This parameter is used in Operator Tab, page 188.
	Set Mode - Override		Supervisor (2)	Privilege to change override mode.This parameter is used in Engineering Tab, page 190.
	Set Mode - Simulation		Engineer (5)	Privilege to change simulation mode. This

Section	Parameter description	Type	Default	Additional remarks
				parameter is used in Engineering Tab, page 190.
	Set Mode - Maintenance		Maintenance (3)	Privilege to change maintenance mode. This parameter is used in Engineering Tab, page 190.
	Set Mode - Out of Service		Supervisor (2)	Privilege to change out of service mode. This parameter is used in Engineering Tab, page 190.
	Set Reset		Operator (Confirmed) (1)	Privilege to reset. This parameter is used in Operator Tab, page 188.
	Set Alarm Acknowledge		Operator (1)	Privilege to acknowledge alarm. This parameter is used in Alarm Tab, page 67.
	Set Bypass Conditions		Supervisor (2)	Privilege to set bypass conditions mode. This parameter is used in Engineering Tab, page 190.
	Set Bypass All Interlocks		Supervisor (2)	Privilege to bypass all interlocks. This parameter is used in Engineering Tab, page 190.
	Set Asset Maintenance		Operator (1)	Privilege to set asset maintenance mode. This parameter is used in Engineering Tab, page 190.
Events	Enable log	`Boolean`	False	– **NOTE:** This parameter is also applicable for local panel.
Historize	Setpoints	`Boolean`	False	It allows to historize the change setpoints.
	Owner/Mode			It allows to historize the change in mode and owner.

Asset and Display Parameters

For more details, refer to the topic describing the parameters of asset and display, page 43.

Alarm Parameters

For more details, refer to the topic describing the parameters of alarm, page 45.

Failures

For more details, refer to the topic describing the parameters of failures, page 74.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Maintenance

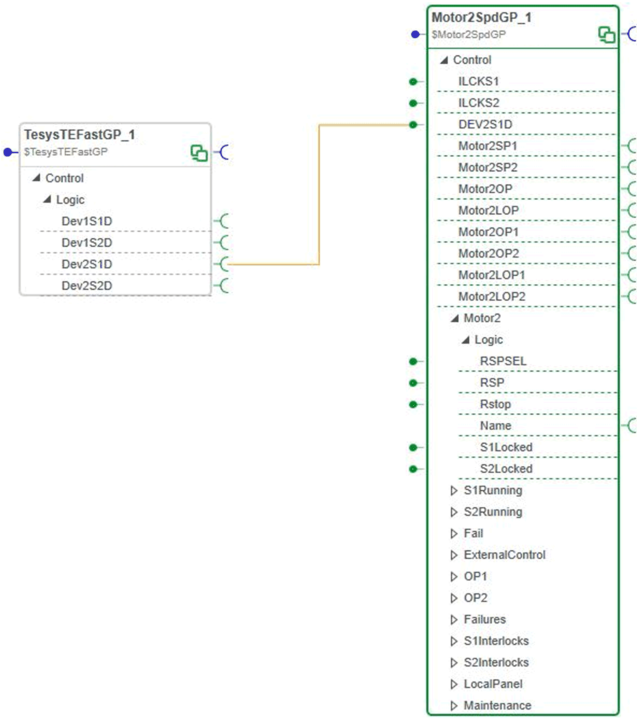
For more details, refer to the topic describing the parameters of maintenance, page 89.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$Motor2SpdGP` (also applicable for $Motor2SpdCE) template as it appears in `Links Editor` and an example of other templates connected to it:



The table describes the control module template interfaces:

Interface identifier	Type/Role	Description	Example
`S1Locked/ S2Locked`	*$Bool/Ref*	Speed lock input interface to 2 direction motor.	A digital input connected to lock the speed S1 and S2 in forward and reverse direction.

For interface identifiers, refer to the control module template interfaces, page 49.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

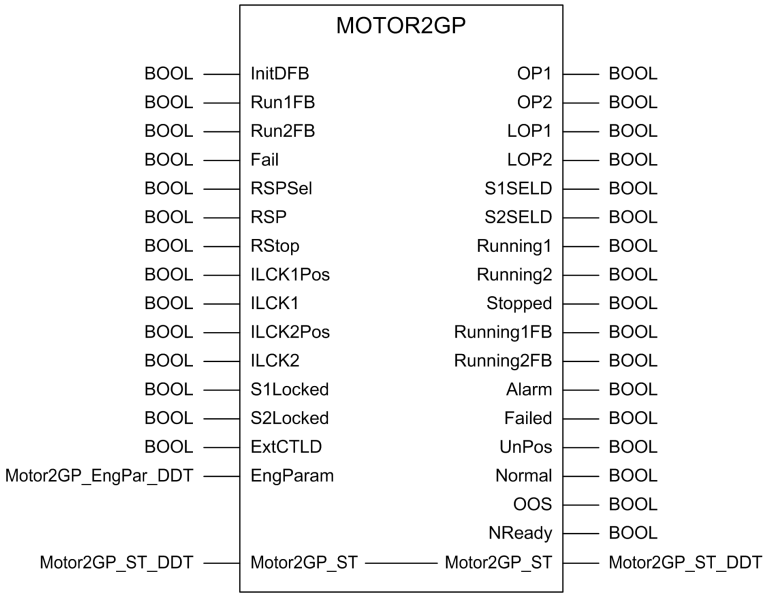
Overview

This section describes the `MOTOR2GP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Name	Type	Description
`InitDFB`	`BOOL`	If a rising edge is detected on this input, the internal timers of the block are reset and detected alarms are set to 0.
`Run1FB`	`BOOL`	1 = Indicates to the DFB that motor is running in forward direction/speed 1. Usually, it is used to connect the digital input of a motor for switching operation confirmation.
`Run2FB`	`BOOL`	1 = Indicates to the DFB that motor is running in reverse direction/speed 2. Usually, it is used to connect the digital input of a motor for switching operation confirmation.
`Fail`	`BOOL`	1 = Indicates to the DFB a detected error in the device. Usually, it is used to connect the digital thermal trip input of a motor switch.
`RSPSel`	`BOOL`	Remote set point selection. 0 = Forward direction/Speed 1 set point. 1 = Reverse direction/Speed 2 set point.
`RSP`	`BOOL`	If `RStop` is connected then the rising edge (Pulse based control) sends the command to start the motor only when program/remote cascade mode is configured.
`RStop`	`BOOL`	A rising edge sends a stop command only when program/remote cascade mode is configured.
`ILCK1Pos`	`BOOL`	Interlock position (1 = Forward direction/speed 1 position will be considered during ILCK1 active).
`ILCK1`	`BOOL`	1 = Interlocks the motor at the defined position. **NOTE:** If the motor requires resetting (`STW.RearmR` = 1), the `OP1` and `OP2` outputs are set to 0 if the `Failed` signals are active regardless of the value of `ILCK1Pos` and will wait for the operator to reset the motor.
`ILCK2Pos`	`BOOL`	Interlock position (1 = Reverse direction/speed 2 position will be considered during ILCK2 active).

Name	Type	Description
ILCK2	BOOL	1 = Interlocks the motor at the defined position. **NOTE:** If the motor requires resetting (STW.RearmR = 1), the OP1 and OP2 outputs are set to 0 if the Failed signals are active regardless of the value of ILCK2Pos and will wait for the operator to reset the motor.
S1Locked	BOOL	1 = Enables to block the activation of the actuator 1 (forward direction/ speed 1) setpoint in case of an OP1 transition from 0 to 1.
S2Locked	BOOL	1 = Enables to block the activation of the actuator 2 (reverse direction/ speed 2) setpoint in case of an OP2 transition from 0 to 1.
ExtCTLD	BOOL	1 = Indicates to the DFB the motor is being controlled from outside the system. Avoids the generation of erroneous confirmation messages while the motor is being controlled from outside the system (for example, from an electrical switch controlled by local push buttons that does not reach the control system). Monitoring of the confirmation alarm ceases while the motor is being externally controlled. The motor detected error continues to be evaluated. **NOTE:** If ExtCTLD =1, the OP1 and OP2 outputs can still be manipulated.
EngParam	Motor2GP_EngPar_DDT	Engineering parameters

Motor2GP_EngPar_DDT Type

Name	Type	Description
TimeOut	TIME	Enables to configure the maximum time that the device takes to reach the defined position. When the output is triggered, if the limit switch is not detected even after the TimeOut time is elapsed. The device is switched to detected alarm state if the limit switch is not detected. When it has a value of 0, monitoring is deactivated.
ScanTime	TIME	The minimum time for which the detected alarm signals are kept active. ScanTime confirms that the supervision system acquires the data for detected alarms that are automatically reset.
ConfRearmEn	BOOL	• 1 = Withdraws the output when a non-confirmed operation is detected (*Alarm* = 1) and requires a mandatory manual resetting once the *Alarm* condition disappears. • 0 = The output remains in its previous state when a non-confirmed operation is detected (*Alarm* = 1) and does not need manual resetting once the *Alarm* condition disappears. Refer the TimeOut parameter and the Motor2GP_ST.STW.Alarm input/output.
FailRearmEn	BOOL	• 1 = A manual resetting is required when the detected error/interlock trip alarm condition disappears. • 0 = No manual resetting is required when the detected error/interlock trip alarm condition disappears.
DisStopAlarm	BOOL	1 = Disable device detected error alarm when motor is not running.

Name	Type	Description
ResetSPFail	BOOL	Reset setpoint in case of detected error condition (1 = Restart is required if motor trips due to detected error or alarm).
TrackSPEn	BOOL	Track set-point in case of that the interlock condition is active. **NOTE:** The *TrackSPEn* parameter enables the interlock trip alarm when the device stopped due to interlock conditions.

Outputs

Output Parameter Description

Name	Type	Description
OP1	BOOL	1 = Forward direction/speed 1 field command signal active. Allows the control signal of the motor to be connected. It always follows the LOP1 command. **NOTE:** During simulation the signal will be deactivated.
OP2	BOOL	1 = Reverse direction/speed 2 field command signal active. Allows the control signal of the motor to be connected. It always follows the LOP2 command. **NOTE:** During simulation the signal will be deactivated.
LOP1	BOOL	1 = Forward direction/speed 1 logical command signal active. The current output (OP1) is continuously copied to this variable, except in the simulation mode. Refer to the LOP1 and LOP2 Calculation, page 176.
LOP2	BOOL	1 = Reverse direction/speed 2 logical command signal active. The current output (OP2) is continuously copied to this variable, except in the simulation mode. Refer to the LOP1 and LOP2 Calculation, page 176.
S1SELD	BOOL	1 = Current device forward direction/speed 1 setpoint and with the command not blocked at the time of OP1 activation (refer to the S1Locked input pin, page 173). Refer to the SP Calculation, page 177.
S2SELD	BOOL	1 = Current device reverse direction/speed 2 setpoint and with the command not blocked at the time of OP2 activation (refer to the S2Locked input pin, page 173). Refer to the SP Calculation, page 177.
Running1	BOOL	1 = The motor has reached the position corresponding to setpoint SP = 1 and SPSel = 0. It is evaluated by considering any detectors. Refer to the Running2 output and Motor2GP_ST.CFGW.SIMMD input/output pin, page 178. Refer to the Running1 and Running2 feedback Calculation, page 177.
Running2	BOOL	1 = The motor has reached the position corresponding to setpoint SP = 1 and SPSel = 1. It is evaluated by considering any detectors. Refer to the Running1 output and Motor2GP_ST.CFGW.SIMMD input/output pin, page 178. Refer to the Running1 and Running2 feedback Calculation, page 177.
Stopped	BOOL	1 = Indicates that the motor is stopped. It is activated when neither the Running1FB nor the Running2FB output signals are active.
Running1FB	BOOL	Forward direction/speed 1 position status. 1 = Forward direction/speed 1 position confirmed. It is independent from the other detector if the Run1FB detector is available.

Name	Type	Description
		Refer to the Running1FB and Running2FB feedback Calculation, page 178.
Running2FB	BOOL	Reverse direction/speed 2 position status. 1 = Reverse direction/speed 2 position confirmed. It is independent from the other detector if the Run2FB detector is available. Refer to the Running1FB and Running2FB feedback Calculation, page 178.
Alarm	BOOL	1 = Forward direction/speed 1 or reverse direction/speed 2 unsuccessful operation has occurred.
Failed	BOOL	1 = Indicates an inoperable device. It is evaluated according to the value of the Fail. If FailRearmEn engineering parameter configured then Failed will be high until Rearm provided. When it is activated, a time period is set so that the signal is maintained for a minimum of ScanTime. 0 = After the minimum activation time has passed, it is deactivated when the Fail input signal is inactive.
UnPos	BOOL	1 = Indicates that there is no way to determine which operation is active (indeterminable operation). For more details refer the below table. Refer to the UnPos feedback Calculation, page 178.
Normal	BOOL	1 = Indicates the motor operating in normal mode.
OOS	BOOL	1 = Indicates the device is out of service.
NReady	BOOL	1 = Indicates the motor not ready for operation.

LOP1 and LOP2 Calculation

The evaluation of the device control depending on the interlock status and if the DFB requires resetting is shown in the following table:

STW.RearmR	SC.ILCKD	LOP1 is calculated as:	LOP2 is calculated as:
ON	-	OFF	OFF
OFF	ON	Refer to the below Interlock calculation table.	
OFF	OFF	S1SELD	S2SELD

Interlock Calculation

OP1 and OP2 in interlock state:

When OP1 = OFF and OP2 = ON					
ILCK1	ILCK1Pos	ILCK2	ILCK2-Pos	LOP1	LOP2
ON	OFF	OFF	OFF	Holds the last output state (Operator can change setpoint to Stop).	
ON	ON	OFF	OFF		
ON	OFF	ON	OFF	OFF	OFF
ON	OFF	ON	ON	OFF	ON
ON	ON	OFF	ON	ON	OFF
ON	ON	ON	ON	Holds the last output state.	
OFF	OFF	ON	OFF	OFF	OFF
OFF	OFF	ON	ON	OFF	ON
OFF	-	OFF	-	Holds the last output state.	

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

The *ConfRearmEn* engineering parameter has to be set to 1 for the *OP* pin of the function block output to become 0, under non-confirmed operations.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: Non-confirmed operations are those for which commands have been given and to which there is no confirming reply from the equipment.

SP Calculation

The signal evaluation depending on the SC.Rem and RSP inputs and the Motor2GP_ST.CFGW.Owner input/output is shown in the following table:

Owner	Rem	S1SELD is calculated as	S2SELD is calculated as
OFF	OFF	Sequential control (not SC.LSPSel) and SC.LSP or Pulse base control (SC.LStart1, SC.LStop)	Sequential control (SC.LSPSel) and SC.LSP) or Pulse base control (SC.LStart2, SC.LStop)
OFF	ON	Continuous control (not RSPSel) and RSP or Pulse base control if RStop pin is connected (not RSPSel, RSP, RStop)	Continuous control (not RSPSel) and RSP) or Pulse base control if RStop pin is connected (RSPSel, RSP, RStop)
ON	-	Not Motor2GP_ST.LSPSel and Motor2GP_ST.LSP	Motor2GP_ST.LSPSel and Motor2GP_ST.LSP

Feedback Calculation for Running1 and Running2

ExtCTLD	Run1FB connected	Run2FB connected	Simulation	Override	Running1 is calculated as:	Running2 is calculated as:
-	-	-	-	ON	LOP1	LOP2
-	-	-	ON	OFF	LOP1 and (TimeOut/2)	LOP2 and (TimeOut/2)
OFF	OFF	OFF	OFF	OFF	LOP1	LOP2
ON	OFF	OFF	OFF	OFF	OFF	OFF
OFF	OFF	ON	OFF	OFF	Run2FB and OP1	Run2FB and OP2
ON	OFF	ON	OFF	OFF	OFF	OFF
OFF	ON	OFF	OFF	OFF	Run1FB and OP1	Run1FB and OP2
ON	ON	OFF	OFF	OFF	OFF	OFF
OFF	ON	ON	OFF	OFF	Run1FB and (not Run2FB)	Run2FB and (not Run1FB)

Feedback Calculation for Running1FB, Running2FB, and UnPos

ExtCTLD	Run1FB connected	Run2FB connected	Simulation	Override	Running1FB is calculated as:	Running2FB is calculated as:	UnPos is calculated as:
-	-	-	-	ON	LOP1	LOP2	OFF
-	-	-	ON	OFF	LOP1 and (TimeOut/2)	LOP2 and (TimeOut/2)	OFF
OFF	OFF	OFF	OFF	OFF	LOP1	LOP2	OFF
ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF
OFF	OFF	ON	OFF	OFF	Run2FB and OP1	Run2FB and OP2	Run2FB and (not OP1 and not OP2)
ON	OFF	ON	OFF	OFF	OFF	OFF	Run2FB
OFF	ON	OFF	OFF	OFF	Run1FB and OP1	Run1FB and OP2	Run1FB and (not OP1 and not OP2)
ON	ON	OFF	OFF	OFF	OFF	OFF	Run1FB
OFF	ON	ON	OFF	OFF	Run1FB	Run2FB	Run1FB and Run2FB

Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
Motor2GP_ST	Motor2GP_ST_DDT	Data structure of the DFB control and status.

Motor2GP_ST_DDT Type

Name	Bit name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	Running1FB	*Bit0*	BOOL	Refer to Running1FB output pin, page 175.
	Running2FB	*Bit1*	BOOL	Refer to Running2FB output pin, page 175.
	Unpos	*Bit2*	BOOL	Refer to Unpos output pin, page 175.
	LOP1	*Bit4*	BOOL	Refer to LOP1 output pin, page 175.
	LOP2	*Bit5*	BOOL	Refer to LOP2 output pin, page 175.
	Rem	*Bit6*	BOOL	Refer to Rem, page 179.
	ILCK	*Bit7*	BOOL	1 = Interlock condition is active (ILCK1 or ILCK2).
	RearmR	*Bit8*	BOOL	Refer to RearmR, page 179.
	ExtCTLD	*Bit9*	BOOL	Refer to ExtCTLD input pin, page 173.
	Alarm1	*Bit10*	BOOL	1 = Forward direction/speed 1 confirmation detected error active.
	Alarm2	*Bit11*	BOOL	1 = Reverse direction/speed 2 confirmation detected error active.
	Failed	*Bit12*	BOOL	Refer to Failed output pin, page 175.
	S1Locked	*Bit14*	BOOL	Refer to S1Locked input pin, page 173.
	S2Locked	*Bit15*	BOOL	Refer to S2Locked input pin, page 173.
STW1	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.

Name	Bit name	Bit	Type	Description
	Nready	*Bit0*	BOOL	Refer to Nready output pin, page 175.
	SP1Locked	*Bit1*	BOOL	Indicates whether the forward direction/speed 1 setpoint change buttons in the monitoring (HMI) system should be locked (1) or not (0).
	SP2Locked	*Bit2*	BOOL	Indicates whether the reverse direction/speed 2 setpoint change buttons in the monitoring (HMI) system should be locked (1) or not (0).
	ILCK1Trip	*Bit3*	BOOL	1 = Interlock 1 trip active.
	ILCK2Trip	*Bit4*	BOOL	1 = Interlock 2 trip active.
	ILCK1	*Bit5*	BOOL	Refer to ILCK1 input pin, page 173.
	ILCK2	*Bit6*	BOOL	Refer to ILCK2 input pin, page 173.
	FailAlarm	*Bit11*	BOOL	Refer to FailAlarm, page 179.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read-write access to the data contained in this word.
	Owner	*Bit0*	BOOL	Read/write access Refer to Owner, page 179.
	ILCKBP	*Bit1*	BOOL	Read/write access 1 = Enables the interlock to be bypassed.
	Override	*Bit2*	BOOL	Refer to Override, page 179.
	Rearm	*Bit3*	BOOL	Write-only access Refer to Rearm, page 179.
	LSPSel	*Bit4*	BOOL	Read/write access Refer to LSPSel, page 179.
	LSP	*Bit5*	BOOL	Read/write access Refer to LSP, page 179.
	SimFBFail	*Bit12*	BOOL	1 = Feedback will not simulate in simulation mode.
	Sim	*Bit13*	BOOL	Read/write access Refer to Sim, page 179.
	Maint	*Bit14*	BOOL	Refer to Maint, page 179.
	OOS	*Bit15*	BOOL	Refer to OOS output pin, page 175.

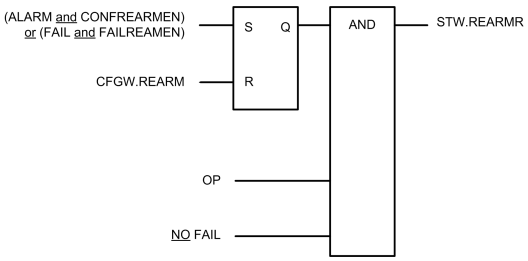
Public Variables

Public Variable Description

Variable	Type	Description
SC	Motor2GP_SC_DDT	Provides the frequently needed data to monitor the device status and to control it from the sequential control.

Name	Type	Description
Rem	BOOL	Read/write access • 1 = Enables the DFB to be configured to remote set-point Rem. • 0 = Enables the DFB to be configured to local set-point LSP.
LSPSel	BOOL	Read/write access 1 = Enables to select the actuator to be affected by the local set-point (0 corresponds to the forward direction/speed 1 and 1 corresponds to the reverse direction/speed 2) usually for the sequential control if the owner is the Program (OWNER is 0) and the selected set-point is Local (Rem is 0). In any other case, the DFB continuously assigns LSPSel one of following value:

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Owner</th><th>Rem</th><th>SC.LSPSel is calculated as:</th></tr>
<tr><td>OFF</td><td>OFF</td><td>SC.LSPSel
(last value or new value assigned from the sequential control)</td></tr>
<tr><td>OFF</td><td>ON</td><td>SPSel
(calculated from the RSPSel input)</td></tr>
<tr><td>ON</td><td>ON</td><td>SPSel
(last value or new value assigned by the Operator)</td></tr>
</table>
</td></tr>
<tr><td>LSP</td><td>BOOL</td><td>
Read/write access
1 = Enables the assignment of the target local set-point usually for the sequential control if the owner is the Program (Owner input/output is OFF) and the selected set-point is Local (SC.Rem public variable is OFF).
0 = The block automatically assigns the current value of the resulting set-point. It is described as follows:
<table>
<tr><th>Owner</th><th>Rem</th><th>SC.LSP</th></tr>
<tr><td>OFF (Program)</td><td>OFF</td><td>SC.LSP
(last value or new value assigned from the sequential control)</td></tr>
<tr><td>OFF</td><td>ON</td><td>SP (calculated from the RSP)</td></tr>
<tr><td>ON (Operator)</td><td>ON</td><td>SP (last value or new value assigned by the Operator)</td></tr>
</table>
</td></tr>
<tr><td>LStart1</td><td>BOOL</td><td>Read/write access
1 = Rising edge to enable forward direction/speed 1 Start command while the owner is the Program and Local.</td></tr>
<tr><td>LStart2</td><td>BOOL</td><td>Read/write access
1 = Rising edge to enable reverse direction/speed 2 Start command while the owner is the Program and Local.</td></tr>
<tr><td>LStop</td><td>BOOL</td><td>Read/write access
1 = Rising edge to enable Stop command while the owner is the Program and Local.</td></tr>
<tr><td>OP1</td><td>BOOL</td><td>Read-only access
Refer to OP1 output pin, page 175.</td></tr>
<tr><td>OP2</td><td>BOOL</td><td>Read-only access
Refer to OP2 output pin, page 175.</td></tr>
<tr><td>LOP1</td><td>BOOL</td><td>Read-only access
Refer to LOP1 output pin, page 175.</td></tr>
<tr><td>LOP2</td><td>BOOL</td><td>Read-only access
Refer to LOP2 output pin, page 175.</td></tr>
<tr><td>S1SELD</td><td>BOOL</td><td>Read-only access
Refer to S1SELD output pin, page 175.</td></tr>
<tr><td>S2SELD</td><td>BOOL</td><td>Read-only access
Refer to S2SELD output pin, page 175.</td></tr>
<tr><td>Running1</td><td>BOOL</td><td>Read-only access
Refer to Running1 output pin, page 175.</td></tr>
<tr><td>Running2</td><td>BOOL</td><td>Read-only access
Refer to Running2 output pin, page 175.</td></tr>
<tr><td>Stopped</td><td>BOOL</td><td>Read-only access
Refer to Stopped output pin, page 175.</td></tr>
<tr><td>Running1FB</td><td>BOOL</td><td>Read-only access
Refer to Running1FB output pin, page 175.</td></tr>
</table>

Name	Type	Description
Running2FB	BOOL	Read-only access Refer to Running2FB output pin, page 175.
Unpos	BOOL	Read-only access Refer to Unpos output pin, page 175.
Alarm1	BOOL	Read-only access Refer to Alarm1 output pin, page 175.
Alarm2	BOOL	Read-only access Refer to Alarm2 output pin, page 175.
Failed	BOOL	Read-only access 1 = Fail input pin is active (Read-only).
FailAlarm	BOOL	Read-only access 1 = Detected error alarm is active.
ILCKD	BOOL	Read-only access The signal evaluation depending on the ILCKD1 and ILCKD2 input and the MotorGP_ST.CFGW.ILCKBP input/output is shown as follows:
ILCKD1	BOOL	Read-only access Forward direction/speed 1 interlock and not bypassed.
ILCKD2	BOOL	Read-only access Reverse direction/speed 2 interlock and not bypassed.
Rearm	BOOL	Read/write access Allows the device to be reset (1). The block sets the signal to 0 after each execution and sets the signal indicating that the resetting is required (CFGW.Rearm) to 0.
RearmR	BOOL	Read/write access Indicates if the device requires resetting (1) after the detection of an inoperable device condition if it has been configured through the ConfRearmEn and/or FailRearmEn inputs. The signal activation logic included in the DFB is as follows. 
S1Locked	BOOL	Read-only access Refer to S1Locked input pin, page 173.
S2Locked	BOOL	Read-only access Refer to S1Locked input pin, page 173.
Owner	BOOL	Read/write access Enables to configure whether the set-point is set by the Program (0) or by the Operator (1).
Normal	BOOL	Read/write access Refer to Normal output pin, page 175.
Override	BOOL	Read/write access Enables to determine if the DFB is operating with external output (1 = Override) or not (0 = Normal).
Sim	BOOL	Read/write access Enables to select the value of the simulated measurement in engineering units.

Name	Type	Description
Maint	BOOL	Read-only access Rising edge to activate maintenance mode (1 = Maintenance mode active).
OOS	BOOL	Read/write access Refer to OOS output pin, page 175.
NReady	BOOL	Read-only access Refer to NReady output pin, page 175.
ExtCTLD	BOOL	Read-only access Refer to ExtCTLD input pin, page 173.

Optional Service - Local Panel (MOTOR2LPGP)

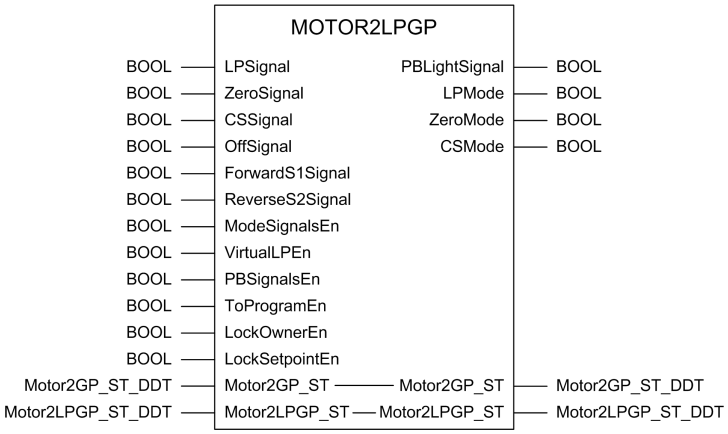
Overview

This chapter describes the optional services - local panel (MOTOR2LPGP) DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Name	Type	Description
LPSignal	BOOL	1 = The local panel mode switch is in the Local Panel position. Refer to the table included with the description of the LPMode output pin, page 184.
ZeroSignal	BOOL	1 = The local panel mode switch is in the zero position.This signal is optional even if a mode switch is available on the local panel. When this signal is a logical high, user is not allowed to operate the device from the faceplate and also from the DFB. Hence, all the functions in the faceplate will be disabled. Refer to the table included with the description of the LPMode output pin, page 184.
CSSignal	BOOL	1 = The local panel mode switch is in the Control System position. Refer to the table included with the description of the LPMode output pin, page 184.

Name	Type	Description
OffSignal	BOOL	1 = The OFF push button on the local panel is pressed. **NOTE:** OFF push button signal has higher priority than ON push button signal.
ForwardS1Signal	BOOL	Rising edge of this signal indicates direct/speed1 push button is pressed.
ReverseS1Signal	BOOL	Rising edge of this signal indicates reverse/speed1 push button is pressed.
ModeSignalsEn	BOOL	1 = Enables the use of the Local Panel/Zero/Control System (or Local Panel - Control System) mode switch on the local panel. Usually configured in development phase based on the characteristics of the local panel being used. If the mode switch is enabled, the LPSignal, ZeroSignal, and CSSignal inputs are considered to determine the operating modes of the local panel. Refer to the table included with the description of the LPMode output pin, page 184. **NOTE:** *ModeSignalsEn* input pin has higher priority than the *VirtualLPEn* input pin signal.
VirtualLPEn	BOOL	1 = This signal enables the operator to select the Local/Control System mode of operation from the HMI, so that the push button signals are enabled for operation. **NOTE:** When the owner is program, the Local/Control System mode selection drop-down list will be visible but disabled for operation however, when the owner is operator the Local Panel drop-down list will be accessible for operation. Refer to the table included with the description of the LPMode output pin, page 184.
PBSignalsEn	BOOL	1 = Enables to accept the OFF and ON push button signals from the local panel. This signal is not applicable if the mode switch is enabled (refer to the ModeSignalsEn input). It means that push button signals are considered when in Local Panel mode. When the mode switch is disabled (ModeSignalsEn = 0), the PBSignalsEn signal enables/disables the push button signals from the control system itself based on the relevant process conditions. The SC.DisableLP public variable is also considered for determining whether the push buttons are enabled or not when *LPMode* pin signal is high, as shown in the following table.

VirtualLPEn	ModeSignalsEn	PBSignal-sEn	SC. DisableLP	Push buttons enabled?
OFF	OFF	OFF	-	NO
OFF	OFF	-	ON	NO
OFF	OFF	ON	OFF	YES
OFF	ON	-	-	YES
ON	OFF	OFF	-	YES

Therefore, this signal can be used to enable/disable the push buttons on a simple local panel that does not feature a mode switch (Local Panel/Zero/Control System) based on relevant process conditions.

ToProgramEn	BOOL	1 = Change to program mode when control is switched to control system mode and VirtualLPEn = 0.
LockOwnerEn	BOOL	1 = Disables the access of changing the owner from HMI while the mode switch is in local panel. The following table shows how the on-off device owner is evaluated based on *ToProgramEn*, *LockOwnerEn* input signals and the local panel operating mode:

Local Panel Mode	ToProgra-mEn	LockOwne-rEn	Owner
Zero mode	-	-	Operator
Switch to Local Panel	-	-	Operator
Local Panel	-	ON	Operator
Switch to Control System	ON	-	Program

In remaining cases, owner of the on/off device ceases to be defined from the Motor2LPGP DFB and it enables to change it from monitoring (HMI) system with the command word in Motor2LPGP_ST.CFGW input/output of the corresponding Motor2LPGP DFB.

In any case, consider that, the device owner that had been set before switching to the local panel mode is not memorized.

LockSetpointEn	BOOL	1 = Disables the access to set point of on/off devices in the monitoring (HMI) system when it is in local panel mode only, and the owner is operator. Therefore, user can activate this signal to lock setpoint operation from the monitoring (HMI) system while on/off device is controlled from the local panel.

Outputs

Output Parameter Description

Name	Type	Description
PBLightSig-nal	BOOL	1 = The push buttons on the local panel are fully operational, that is, the local panel is in Local Panel mode (either because the mode is selected with the corresponding switch or because the push buttons are enabled in the event that there is no selector switch available or the *VirtualLPEn* is high with the local panel mode enabled).
LPMode	BOOL	1 = The local panel mode is in Local Panel. The following table shows how the local panel operating mode is determined based on the ModeSignalsEn, VirtualLPEn, PBSignalsEn, LPSignal, CSSignal and ZeroSignal, input signals:

Mode-Signal-sEn	Vir-tualLPEn	PBSigna-lsEn	LPSignal	CSSignal	ZeroSignal	Output
0	0	0	0	0	0	CS mode
1	0	0	0	0	0	Zero mode
1	0	0	1	0	0	LP mode and PBLightSignal
1	0	0	0	1	0	CS mode
0	1	1	NA	NA	NA	CS mode
0	1	0	NA	NA	NA	CS mode
0	0	1	0	0	0	LP mode and PBLightSignal
1	1	0	1	0	0	LP mode and PBLightSignal
1	1	0	0	0	0	Zero mode
1	1	0	0	0	1	Zero mode
1	1	0	0	1	0	CS mode
1	0	1	0	1	0	CS mode
1	1	1	0	1	0	CS mode
1	0	0	0	0	1	Zero mode

ZeroMode	BOOL	1 = The local panel mode is in Zero. User is not allowed to operate the device from the faceplate and also from DFB. Hence, all the functions in the faceplate will be disabled. Refer to the table included with the description of the LPMode output.
CSMode	BOOL	1 = The local panel mode is in control system. Refer to the table included with the description of the LPMode output.

Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
Motor2GP_ST	Motor2GP_ST_DDT, page 178	Data structure corresponding to the 2-speed/2-rotation direction motor is to be controlled from the local panel (for detailed information regarding this structure, refer to the Motor2GP DFB, page 173. The structure provides the information needed for the operation of the MOTOR2LPGP DFB and enables this DFB to control the owner of the 2-speed/2-rotation direction motor.
Mo-tor2LPGP_ST	Motor2LPGP_ST_DDT	Data structure that is used as an interface with the monitoring (HMI) system.

Motor2LPGP_ST.DDT Word Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	LPMode	*Bit0*	BOOL	Refer to the LPMode output pin, page 184.
	ZeroMode	*Bit1*	BOOL	Refer to the ZeroMode output pin, page 184.
	CSMode	*Bit2*	BOOL	Refer to the CSMode output pin, page 184.
	OwnerLocked	*Bit3*	BOOL	Indicates whether the owner change buttons in the monitoring (HMI) system needs to be locked (1) or not (0). Is activated when: • The Local Panel mode is set to Zero or • It is set to Local Panel and the LockOwnerEn input signal has been configured as active (1).
	SPLocked	*Bit4*	BOOL	Indicates whether the set-point change buttons in monitoring (HMI) system needs to be locked (1) or not (0). Is activated when: • The Local Panel mode is set to Zero or • It is set to Local Panel and the LockSetpointEn input signal has been configured as active (1).
	ModeSignalsEn	*Bit5*	BOOL	Refer to the ModeSignalsEn input pin, page 182.
	PBEnabled	*Bit6*	BOOL	Indicates whether the push buttons of the local panel are enabled (1) or not (0). The signal is activated when at least one of the ModeSignalsEn and PBSignalsEn signal inputs is active.
	VirtualLPEn	*Bit7*	BOOL	Refer to the VirtualLPEn input pin, page 182.
STW	LPSignal	*Bit8*	BOOL	Refer to the LPSignal input pin, page 182.
	ZeroSignal	*Bit9*	BOOL	Refer to the ZeroSignal input pin, page 182.
	CSSignal	*Bit10*	BOOL	Refer to the CSSignal input pin, page 182.
	OffSignal	*Bit11*	BOOL	Refer to the OffSignal input pin, page 182.
	ForwardS1Signal	*Bit12*	BOOL	Refer to the ForwardS1Signal input pin, page 182.
	ReverseS2Signal	*Bit13*	BOOL	Refer to the ReverseS2Signal input pin, page 182.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read-only access to the data contained in this word.
	LPEnabled	*Bit0*	BOOL	Read/write access • 1 = Enables the Local Panel mode. • 0 = Enables the Control System mode.

Public Variables

Public Variable Description

Variable	Type	Description
SC	Motor2LPGP_SC_DDT	Provides the frequently needed data for monitoring (HMI) and controlling the DFB.

Motor2LPGP_SC_DDT Type

Name	Type	Description
LPMode	BOOL	Read-only access Refer to the LPMode output pin, page 184.
ZeroMode	BOOL	Read-only access Refer to the ZeroMode output pin, page 184.
CSMode	BOOL	Read-only access Refer to the CSMode output pin, page 184.
DisableLP	BOOL	Read/write access 1 = Disable push button selection of local panel (Read/Write).

Supervision

Overview

This chapter describes the Supervision resources and runtime services that are available for the dual direction motor (motor2) and dual speed motor (motor2S).

Genies

Genie Properties

Refer to Genie Properties, page 62

Genie Icons

Refer to Genie Icons, page 63

Representation

The table describes the genies of the gpl_motor2 and gpl_motor2S library:

Graphic symbol	Genie name	Description
Dual direction motor		
	Motor_Direction	Motor with two rotational directions.
	Motor_Direction_Up	Motor with two rotational and up directions.
Dual speed motor		

Graphic symbol	Genie name	Description
	Motor_Speed	Motor with two speed.
	Motor_Speed_Up	Motor with two speed up directions.

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

Available Tabs

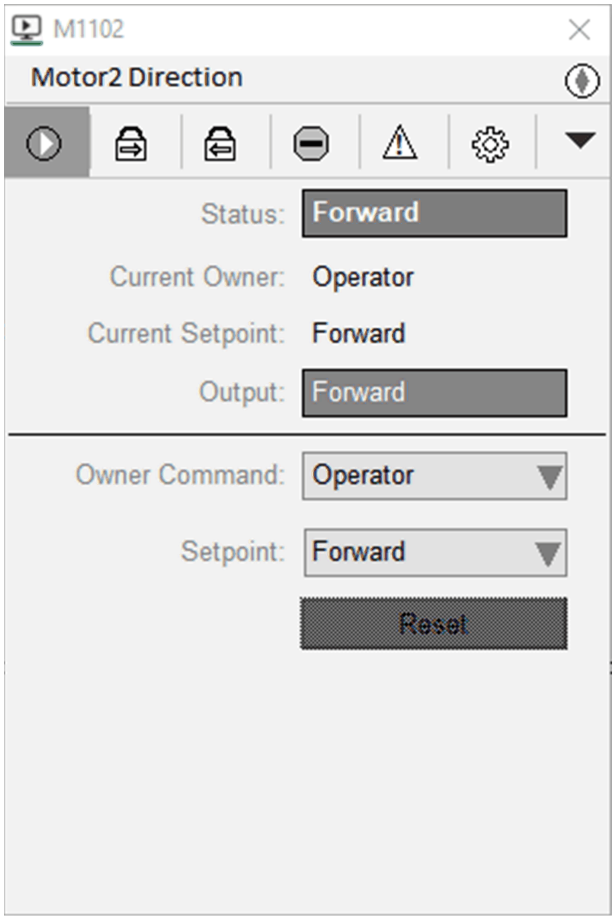
During operation, clicking a motor2/motor2s genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
  - Event, page 66
- Tabs for optional functions, which appear only if configured:
  - Interlock, page 87

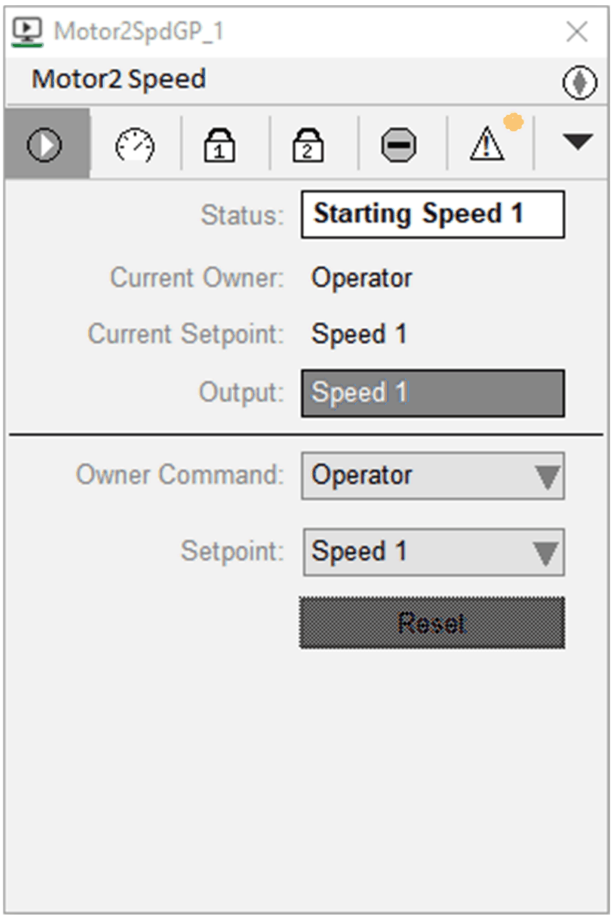
    NOTE: Motor2 can have two interlock tabs, forward/speed1 interlock and reverse/speed2 interlock.
  - Failure, page 78
  - Measures, page 70

Operator Tab

The figure shows an operator tab of **dual direction motor**:



The figure shows an operator tab of **dual speed motor**:

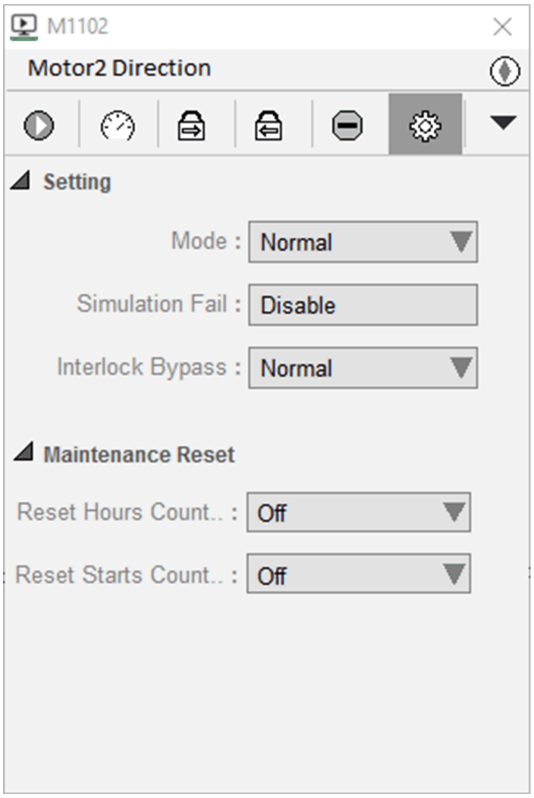


The table describes the control function of an operator tab:

Label	Item name	Description	Security parameter, page 141
Status	EquipStatus	Equipment status of motor2/motor2s. For more details, refer to EquipStatus calculated variable, page 159.	—
Current Owner	Owner	Current owner of motor2/motor2s. For more details, refer to operating owner, page 58.	
Current Setpoint	CurrentSetpoint	Current setpoint of motor2.	
Output	EquipState	Output status of motor2/motor2s. For more details, refer to EquipState calculated variable, page 159.	
OwnerCom-mand	OwnerCommand	Owner command of motor2/motor2s (Program, Operator or Local Panel). **NOTE:** Local panel owner is enabled when the virtual panel is selected.	Set owner
Setpoint	OperatorSet-point	Allows you to change the setpoint, if owner = operator.	Set setpoint
–	Rearm	Allows you to reset manually after the detection of an inoperable device condition or not confirmed operation. **NOTE:** When the control module is reset, the current setpoint shown in operator tab is effective. Also, when you click **Reset**, a dialog box opens to confirm.	Set rearm

Engineering Tab

The figure shows an engineering tab:



The table describes the control function of engineering tab:

Label	Item name	Description	Security parameter, page 165
Setting			
Mode	Mode	Allows you to view and change the mode. For more details, refer to operating mode, page 59.	Set mode - normal, set mode - override, set mode - simulation, set mode - maintenance, and Set mode - out of service **NOTE:** Each configurable mode has the different security parameters
Simulation Fail	SimulationFailActive	Enable or disable feedback simulation. **NOTE:** The Simulation Fail will only be available when the parameter `SimulationFail` is created in the citect parameters. To create, go to **Setup** > **Parameters**. Enter the `GPL.Mode` in **Section Name**, `SimulationFail` in **Name** and set the **Value** = 1.	Set mode - simulation
Interlock Bypass	InterlockBypass	Interlock bypass status Allows you to bypass interlock.	Set bypass conditions, set bypass all interlocks
Maintenance Reset			
Reset Hours Count	MotMntResetHour-sCounter	Reset the hours counter.	Set reset
Reset Starts Count	MotMntResetManeuver-sCounter	Reset the operations counter.	Set reset

Items

Overview

This section describes the variables, equipstatus and equipstate variable, abnormal and action required conditions, alarms, events, and trends items of `dual speed and dual direction motor`.

Variables

The table describes the variable items for the `dual direction motor (motor2) and dual speed motor (motor2S)` that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`FeedbackForward`[1]	Forward feedback	Running Forward/Not Running Forward	*Bit0*
`FeedbackReverse`[1]	Reverse feedback	Running Reverse/Not Running Reverse	*Bit1*
`FeedbackSpeed1`[2]	Speed 1 feedback	–	*Bit0*
`FeedbackSpeed2`[2]	Speed 2 feedback		*Bit1*
`UnknownPosition`	Position feedback detected fault	Unknown Position/Known Position	*Bit2*
`OutputForward`[1]	Forward output status	On/Off	*Bit4*
`OutputReverse`[1]	Reverse output status		*Bit5*
`OutputSpeed1`[2]	Speed 1 output status	On/Off	*Bit4*
`OutputSpeed2`[2]	Speed 2 output status		*Bit5*
`ProgramSetpoint-Mode`	Setpoint source for program owner	Remote/ Sequence	*Bit6*
`InterlockActive`	Interlocked status	Interlock/Healthy	*Bit7*
`RequireRearm`	Reset required	Required/Not Required	*Bit8*
`OwnerExternal`	External owner status	External/Not External	*Bit9*
`AlarmActive`	Forward feedback confirmation detected fault	Active/Inactive	*Bit10*
`FailActive`	Motor forward detected fault		*Bit12*
`ForwardLock`[1]	Forward direction lock status	Locked/Unlocked	*Bit14*
`ReverseLock`[1]	Reverse direction lock status		*Bit15*
`Speed1Lock`[2]	Speed 1 lock status	Locked/Unlocked	*Bit14*
`Speed2Lock`[2]	Speed 2 lock status		*Bit15*
Bit items derived from StatusWord1.			
`NotReady`	Not ready to function	–	*Bit0*
`Setpoint1LockAc-tive`[1]	Forward direction setpoint change is denied		*Bit1*
`Setpoint1LockAc-tive`[2]	Speed 1 setpoint change is denied		*Bit1*
`Setpoint2LockAc-tive`[1]	Backward direction setpoint change is denied		*Bit2*

Item name	Description	Enumeration	Address
Setpoint2LockActive[2]	Speed 2 setpoint change is denied		*Bit2*
Interlock1Trip[1]	Forward direction interlocked trip is active		*Bit3*
Interlock1Trip[2]	Speed 1 interlocked trip is active		*Bit3*
Interlock2Trip[1]	Backward direction interlocked trip is active		*Bit4*
Interlock2Trip[2]	Speed 2 interlocked trip is active		*Bit4*
Interlock1Active	Interlocked trip is active		*Bit5*
Interlock2Active	Interlocked trip is active		*Bit6*
FailAlarmActive	FailAlarm is active		*Bit11*
Bit items derived from ConfigurationWord.			
OwnerSelect	Owner of control module	Operator/ Program	*Bit0*
InterlockBypass	Interlock bypass command	Bypassed/Normal	*Bit1*
InputSource	Input source selection	Operator/Field	*Bit2*
Rearm	Reset detected fault condition	Active/Inactive	*Bit3*
DirectionSetpoint	Operator direction command	Reverse/Forward	*Bit4*
CommandSetpoint	Operator setpoint	On/Off	*Bit5*
SimulationFailActive	Feedback detected failure in simulation mode	–	*Bit12*
Simulation	Simulation mode		*Bit13*
Maintenance	Maintenance mode		*Bit14*
OutOfService	Motor unavailable	Out Of Service/In Service	*Bit15*
IO Items			
StatusWord	Status word	–	MOTOR2GP_ST.STW
StatusWord1	Status word 1	–	MOTOR2GP_ST.STW1
ConfigurationWord	Configuration word		MOTOR2GP_ST.CFGW
Calculated Items			
EquipStatus[1]	Current status of the motor2 For more details, refer to EquipStatus calculated variable, page 194.	–	GPL_Motor2-Status
EquipStatus[2]	Current status of the motor2S For more details, refer to EquipStatus calculated variable, page 194.		GPL_Motor2S-Status
EquipState[1]	Output information For more details, refer to EquipState calculated variable, page 194.		GPL_Motor2E-quipState
EquipState[2]	Output information For more details, refer to EquipState calculated variable, page 194.		GPL_Motor2SE-quipState
Owner	Current owner of the motor2/ motor2s For more details, refer to operating owner, page 58.		GPL_OwnerEx-ternalLo-cal

Item name	Description	Enumeration	Address
Mode	Current mode of the motor2/motor2s For more details, refer to mode function, page 59.		GPL_Mode
OperatorSetpoint[1]	Operator setpoint		GPL_Motor2OperatorSetpoint
OperatorSetpoint[2]	Operator setpoint		GPL_Motor2SOperatorSetpoint
Abnormal	Shows abnormal condition of the motor2/motor2s Refer to abnormal conditions, page 195.		GPL_Abnormal
ActionRequired	Shows action required condition of the motor2/motor2s For more details, refer to action required conditions, page 195.		GPL_ActionRequired
Optional Items			
Local panel			
Bit items derived from LPStatusWord.			
LocalPanelMode	Control from field	Enabled/Disabled	*Bit0*
ZeroMode	Motor2/motor2S control disabled		*Bit1*
ControlSystemMode	Control from remote		*Bit2*
LockedOwner	Owner change disabled from faceplate	Locked/Unlocked	*Bit3*
LockedSetpoint	Setpoint change disabled from faceplate		*Bit4*
LocalPanelEnabled	Control from field	Enabled/Disabled	*Bit5*
PushButtonEnabled	Local panel push buttons		*Bit6*
VirtualLocalPanelEnabled	Local panel enable from control		*Bit7*
LocalPanelSelected	Control from field	Selected/Not Selected	*Bit8*
ZeroSelected	Control disabled		*Bit9*
ControlSystemSelected	Control from remote		*Bit10*
PushButtonStop	Stop push button signal	On/Off	*Bit11*
PushButtonForward	Forward push button signal	On/Off	*Bit12*
PushButtonReverse	Reverse push button signal		*Bit13*
Bit items derived from LPConfigurationWord.			
VirtualSwitch	Local panel selection from faceplate	Local/Control System	*Bit0*
IO Items			
LPStatusWord	Local panel status word	–	Motor2LPGP_ST.STW
LPConfigurationWord	Local panel configuration word		Motor2LPGP_ST.CFGW
Failure	Refer to the topic describing the variable items of failures, page 61.		
Interlock	Refer to the topic describing the variable items of interlocks, page 60.		

Item name	Description	Enumeration	Address
Maintenance	Refer to the topic describing the variable items of maintenance, page 59.		
[1] Applicable for motor2. [2] Applicable for motor2S.			

EquipStatus Calculated Variable

The table describes the *EquipStatus* calculated variable:

Equipment Status	Condition Calculation	Enumeration
–	`UnknownPos`	Unknown
Motor2 Transition[1]	Not `FwdCmd` x Not `RevCmd` x (`FwdRun` + `RevRun`)	Stopping
Motor2 Stopped[1]	Not `FwdCmd` x Not `RevCmd` x Not `FwdRun` x Not `RevRun`	Stopped
Motor2S Transition[2]	Not `Speed1Cmd` x Not `Speed2Cmd` x (`Speed1FB` + `Speed2FB`)	Stopping
Motor2S Stopped[2]	Not `Speed1Cmd` x Not `Speed2Cmd` x Not `Speed1FB` x Not `Speed2FB`	Stopped
Motor2 Running[1]	`FwdCmd` x `FwdRun` x Not `RevRun`	Forward Running
	`RevCmd` x Not `FwdRun` x `RevCmd`	Reverse Running
Motor2 Transition[1]	`FwdCmd` x Not `FwdRun`	Starting Forward
	`RevCmd` x Not `RevCmd`	Starting Reverse
Motor2S Running[2]	`Speed1Cmd` x `Speed1FB` x Not `Speed2FB`	Speed1 Running
	`Speed2Cmd` x Not `Speed1FB` x `Speed2FB`	Speed2 Running
Motor2S Transition[2]	`Speed1Cmd` x Not `Speed1FB`	Starting Speed1
	`Speed2Cmd` x Not `Speed2FB`	Starting Speed2
[1] Applicable for dual direction motor. [2] Applicable for dual speed motor.		

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	Conditions	Enumeration
Passive with Out of Service	`OOS`	Off
Passive to Active1 or Active2 to Active1[1]	Not `OOS` x `Maint` x `FwdCmd`	Forward
Passive to Active2 or Active1 to Active2[1]	Not `OOS` x `Maint` x `RevCmd`	Reverse
Passive to Active1 or Active2 to Active1[2]	Not `OOS` x `Maint` x `Speed1Cmd`	Speed1
Passive to Active2 or Active1 to Active2[2]	Not `OOS` x `Maint` x `Speed2Cmd`	Speed2
Passive	Not `OOS` x `Maint` x Not `FwdCmd` x Not `RevCmd`	Off
Fail	Not `OOS` x Not `Maint` x `Fail`	Failed
Active1 with Interlock[1]	Not `OOS` x Not `Maint` x Not `Fail` x `FwdCmd` x `FwdILCK`	Forward Interlocked
Active1[1]	Not `OOS` x Not `Maint` x Not `Fail` x `FwdCmd` x Not `FwdILCK`	Forward
Active2 with Interlock[1]	Not `OOS` x Not `Maint` x Not `Fail` x `RevCmd` x `RevILCK`	Reverse Interlocked
Active2[1]	Not `OOS` x Not `Maint` x Not `Fail` x `RevCmd` x Not `RevILCK`	Reverse
Active1 with Interlock[2]	Not `OOS` x Not `Maint` x Not `Fail` x `Speed1Cmd` x `Speed1ILCK`	Speed1 Interlocked

Equipment State	Conditions	Enumeration
Active1[2]	Not `OOS` x Not `Maint` x Not `Fail` x `Speed1Cmd` x Not `Speed1ILCK`	Speed1
Active2 with Interlock[2]	Not `OOS` x Not `Maint` x Not `Fail` x `Speed2Cmd` x `Speed2ILCK`	Speed2 Interlocked
Active2[2]	Not `OOS` x Not `Maint` x Not `Fail` x `Speed2Cmd` x Not `Speed2ILCK`	Speed2
Passive with Interlock	Not `OOS` x Not `Maint` x Not `Fail` x Not `FwdCmd` x Not `RevCmd` x (`FwdILCK` + `RevILCK`)	Interlocked
Passive	Not `OOS` x Not `Maint` x Not `Fail` x Not `FwdCmd` x Not `RevCmd` x Not `FwdILCK` x Not `RevILCK`	Off
[1] Applicable for dual direction motor. [2] Applicable for dual speed motor.		

Abnormal Conditions

The table describes the abnormal conditions of motor2 and motor2S:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
`AbnormalOwner`	Abnormal owner status appears, if the current owner is not a normal owner which configured at the instance level.	*Bit0*
`AbnormalMode`	Abnormal mode status appears, if the current operating mode is not normal mode.	*Bit1*
`AbnormalInterlockBypass`	Abnormal interlock bypass condition appears, if the Interlock bypass is enabled.	*Bit2*
`AbnormalOperatorTab`	Abnormal status of operator tab. This condition appears, if the Interlock bypass is enabled.	*Bit8*
`AbnormalFailTab`	Abnormal status of failure tab.	*Bit10*
`AbnormalEngineeringTab`	Abnormal status of engineering tab.	*Bit12*
`AbnormalDirection1Interlock-Tab`	Abnormal status of direction 1 interlock tab.	*Bit13*
`AbnormalDirection2Interlock-Tab`	Abnormal status of direction 2 interlock tab.	*Bit14*

Action Required Conditions

The table describes the action required conditions of motor2 and motor2S:

Item name	Description	Address
Bit items derived from ActionRequired variable word.		
`ActionOperatorTab`	Action required status of operator tab.	*Bit8*
`ActionFailTab`	Action required status of the failure tab.	*Bit10*
`ActionDirection1Interlock-Tab`	Action status of direction 1 interlock tab.	*Bit13*
`ActionDirection2Interlock-Tab`	Action status of direction 2 interlock tab.	*Bit14*

Alarms

The table describes the advanced alarm items that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
MotMntOperationHour-sAlm	Running hours counter overflow	GPL_BitValueCheck (StatusWord, 0)
MotMntManeuversNumber-sAlm	Operations counter overflow	GPL_BitValueCheck (StatusWord, 1)
Alarm	Motor2/Motor2S inoperable	GPL_BitValueCheck (StatusWord, 10,11)
Fail	Confirmation detected failure	GPL_BitValueCheck (StatusWord1,11)
InterlockTrip	Interlock trip	GPL_BitValueCheck (StatusWord1, 3, 4)

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
StateAlarm	All changes in the motor2/motor2S status and output.
ModeAlarm	All changes in the mode and owner.
ActionAlarm	All actions by the operator.

Trends

No trend item is used by Supervision components for motor2 and motor2S .

Variable Speed Motor

What's in This Chapter

Template 197
Control 203
Optional Service - Local Panel (MOTORVSLPGP) 212
Supervision 218

Overview

This chapter describes the components that provide the functionality of template, control services and Supervision functions of the variable speed motor.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$MotorVSGP` template.

Description

General Description

The `$MotorVSGP` process control module template is used to manage motors with a variable speed drive, independently if the speed drive is switched through communication, I/O wiring, or a mix of both.

The process template is designed to allow managing the associated devices from the sequential control, the continuous control, and/or the Supervision system depending on their configuration and on the system needs.

Function Description

The main functions of the template are described in the following table:

Function	Description
Owner selection	The object manages control system level which is the owner (operator or program). As a result, it is responsible for setting the setpoint for the desired position.
Local and remote setpoint	Enables to control the object with a setpoint that is determined through a set-point selector (local or remote). The local setpoint is assigned to the sequential control or to the command received from the Supervision system while the remote setpoint is assigned to the DFB control from the logic implemented in the continuous control.
Running Status	Allows to determine the running status of the element to be controlled and monitored with the help of Run and Stop feedback.
Interlocking	The template gives a command to the device to move to the defined position in case of an active interlock. An interlock bypass and Setpoint reset in case of interlock function is available.
Multispeed	Enables to select between speed configurable through an analog signal or between 3 preset speeds in the speed drive.
Mode selection	The object can be switched to different modes of operation like Simulation, Maintenance, Out Of Service and Override.

Function	Description
Inoperable device	Allows to monitor inoperable device condition (optional). Template can be configured to withdraw output based on this condition and setpoint can be withdrawn in case of detected error. Commonly used for detecting whether or not the motor switch has tripped due to thermal overload.
Detection of unsuccessful operation	If position is unsuccessful within a configured time after the output command is activated, DFB detects unsuccessful operation.
Manual resetting	Allows you to reset manually after the detection of an inoperable device condition or unsuccessful operation. After reset, device output will follow set point. It also allows to reset all the rearm conditions of optional services as well on specific configuration.

Optional Services

Available optional services are:

- **Interlocks**
- **Failures**
- **Local Panel**
- **Maintenance**

Parameters

Parameters

The `$MotorVSGP` template provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default value	Additional remarks
Configuration	**Enable reverse direction** **Enable analog speed setpoint**	`Enum`	*Refinement (3)*	Allows enabling or disabling the reverse direction and analog speed setpoint: • **TRUE**: Enables reverse direction/enables analog speed setpoint of motor by setting value of the corresponding pin to TRUE. • **FALSE**: Disables reverse direction/ disables analog speed setpoint of motor by setting value of the corresponding pin to FALSE. • **Interface**: Enables the corresponding interface, allowing you to provide the value through an interface of type $Bool. • **Refinement**: Allows you to set the value during refinement. The default value that is generated for the corresponding pin is NULL.
Setpoint	**Reset on failure**	`Boolean`	*FALSE*	Resets the setpoint when the motor is non operational.

Section	Parameter description	Type	Default value	Additional remarks
	Track setpoint on interlock	Boolean	*TRUE*	The interlock setpoint is tracked by the equipment setpoint when the interlock condition is active.
	Force remote setpoint	Boolean	*FALSE*	An option to control the motor always remotely with any configuration.
	Force analog setpoint	Boolean	*FALSE*	An option to control the motor using analog setpoint value.
Failure	**Confirmation timeout**	Duration	*00:00:03*	Maximum time for feedback confirmation.
	Minimum alarm duration		*00:00:05*	Minimum time to maintain the confirmation alarm or detected error alarm.
	Disable alarm when motor is not running	Boolean	*FALSE*	It disable the alarms in case of non operational motor.
Rearm	**Required for failure**	Boolean	*TRUE*	Device requires a reset after device error or interlock trip alarm.
	Required for confirmation failure			Device requires a reset after confirmation error.
	Rearm resets conditions		*FALSE*	

Inputs (Digital Input)

The table describes the digital input:

Name	Description
Running	It allows to connect the forward direction feedback of the motor.
Fail	It allows to connect the detected error feedback of the motor.
PowerRemoval	It allows to connect the detected error feedback of the motor.
ExternalControl	It allows to control the device externally.
RSPSel	It allows to connect the remote set-point select condition of the valve.

For more details, refer to the topic describing the parameters of digital inputs, page 44.

Outputs (Digital Output)

The table describes the digital output:

Name	Description
OP	It allows to connect the motor output signal to the output channel.
Direction	–

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Failures

For more details, refer to the topic describing the parameters of failures, page 74.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Maintenance

For more details, refer to the topic describing the parameters of maintenance, page 89.

Local Panel (Logic)

For more details, refer to the topic describing the parameters of logic, page 45.

Local Panel (Digital Input)

The table describes the digital input:

Name	Description
LocalMode	Local panel selection signal from mode selector switch.
ZeroMode	Zero/Off selection signal from mode selector switch.
CSDISignal	Control selection signal from selector switch.
OFFPB	Motor start signal from local panel.
ForwordPB	Forward push button from local panel.
ReversePB	Reverse push button from local panel.
IncSPPB	Increment setpoint push button from local panel.
DecSPPB	Decrement setpoint push button from local panel.
SelectS1	Select S1 speed push button from local panel.
SelectS2	Select S2 speed push button from local panel.
SelectS3	Select S3 speed push button from local panel.

For more details, refer to the topic describing the parameters of digital inputs, page 44.

Local Panel (Digital output)

The table describes the digital output:

Name	Description
PBLight	Output signal to indicate that push buttons are operational.

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default value	Additional remarks
Operation	**Normal owner**	Enum	*Program and Cascade (5)*	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	**Set owner**	Enum	*Operator (1)*	Privilege to change owner. This parameter is used in Operator Tab, page 220.
	Set setpoint		*Operator (1)*	Privilege to change setpoint. This parameter is used in Operator Tab, page 220.
	Set mode - override		*Supervisor (2)*	Privilege to change override mode.This parameter is used in Engineering Tab, page 221.
	Set mode - simulation		*Engineer (5)*	Privilege to change simulation mode. This parameter is used in Engineering Tab, page 221.

Section	Parameter description	Type	Default value	Additional remarks
	Set mode - maintenance		*Maintenance (3)*	Privilege to change maintenance mode. This parameter is used in Engineering Tab, page 221.
	Set mode - out of service		*Supervisor (2)*	Privilege to change out of service mode. This parameter is used in Engineering Tab, page 221.
	Set reset		*Operator (Confirmed) (10)*	Privilege to reset. This parameter is used in Operator Tab, page 220.
	Set alarm acknowledge		*Operator (1)*	Privilege to acknowledge alarm. This parameter is used in Alarm Tab, page 67.
	Set bypass conditions		*Supervisor (2)*	Privilege to set bypass conditions mode. This parameter is used in Engineering Tab, page 221.
	Set bypass all interlocks		*Supervisor (2)*	Privilege to bypass all interlocks. This parameter is used in Engineering Tab, page 221.
Trend	For more details, refer to the topic describing the parameters of trend, page 47.			
Variable Tag Disable	**Disable speed 1**	`Boolean`	*FALSE*	–
	Disable speed 2			–
	Disable speed 3			–
Events	**Enable log**	`Boolean`	*FALSE*	–
Historize	**Present value**	`Boolean`	*FALSE*	–
	Setpoint			It allows to historize the change setpoints.
	Setpoint selection			It allows to historize the change setpoint selection.
	Speed 1			–
	Speed 2			–
	Speed 3			–
	Enable historize output			–

Asset and Display Parameters

For more details, refer to the topic describing the parameters of asset and display, page 43.

Alarm Parameters

For more details, refer to the topic describing the parameters of alarm, page 45.

Failures

For more details, refer to the topic describing the parameters of failures, page 74.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Maintenance

For more details, refer to the topic describing the parameters of maintenance, page 89.

Local Panel

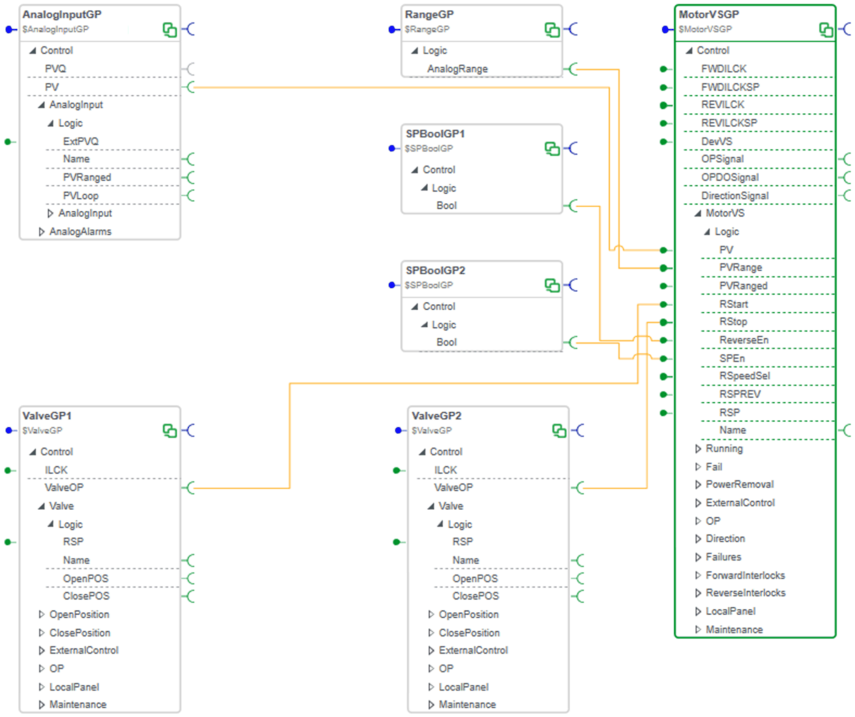
For more details, refer to the topic describing the parameters of local panel, page 44.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

The following figure depicts an example of an instance of `$MotorVSGP` (also applicable for `$MotorVSCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



The following table lists the link interface that creates link symbol on faceplate:

Interface name	Runtime parameter
`RSP`	*SetpointNavigate*
`PV`	*PVNavigate*
`PVRange`	
`PVRanged`	

For more details, refer to Supervision chapter > Faceplate > Navigation Link of General Purpose Library User Guide.

Refer to common control module template interface, page 47 for more details on common interfaces.

The table describes the control module template interfaces:

Interface identifier	Type/Role	Description	Example
`SPEn`	*$Bool*	Setpoint enable input interface from other template.	A boolean type of interface can be connected.

Control

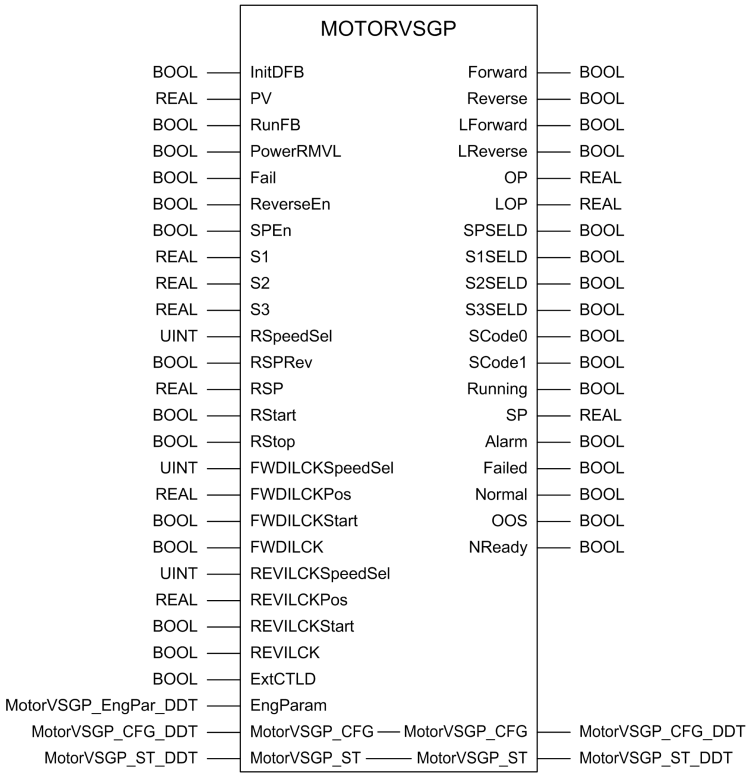
Overview

This section describes the `MOTORVSGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
`InitDFB`	`BOOL`	If a rising edge is detected on this input, the internal timers of the block are reset and detected alarms are set to 0.
`PV`	`REAL`	Current value of the speed measurement. Comes from the speed drive or from a measurement device.
`RunFB`	`BOOL`	1 = Indicates to the DFB that motor is running in forward direction/reverse direction. Usually, it is used to connect the digital input of a motor for switching operation confirmation.
`PowerRMVL`	`BOOL`	1 = Indicates to the DFB that the speed drive is disabled (Power Removal). After activating this signal, the system requires resetting. `MotorVSGPST.CFGW.Rearm` = 1. If you use the `PowerRMVL` signal of the speed drive without requiring resetting wire it to a `ILCKONGP`/`ILCKOFFGP` DFB input that does not require resetting and the output of the latter to the `ILCK` input.

Parameter	Type	Description
Fail	BOOL	1 = Indicates to the DFB an error is detected in the device. Usually, it is used to connect the digital thermal trip input of a motor switch.
ReverseEn	BOOL	1 = Enable to configure whether the speed drive can operate in forward and reverse rotation 0 = Enable to configure whether the speed drive can operate in only forward.
SPEn	BOOL	1 = Allows enabling the analog speed set-point (SP — analog setpoint). Applies to the operator/local/remote owner.
S1 S2 S3	REAL	Enables (1) the availability of the multispeed setpoint x (value of the signal other than 0) and reports the speed configured on the speed drive as Sx to the Supervision system.
RSpeedSel	UINT	Enables to select which remote setpoint should be considered in the rogram/remote mode. The remote set-point is calculated as follows: **RSpeedSel** — **The remote setpoint is:** 0 — RSP 1 — Speed 1 (S1 – configured in the speed drive) 2 — Speed 2 (S2 – configured in the speed drive) 3 — Speed 3 (S3 – configured in the speed drive) **NOTE:** If the analog setpoint is selected (xSpeedSel = 0) and the signal has not been enabled (SPEn = 0), the setpoint indicated by S1 will be considered as the active setpoint regardless of its state.
RSPRev	BOOL	1 = Selects rotation in reverse direction for the remote setpoint in program/remote Owner. 0 = 1 = Selects rotation in forward direction for the remote setpoint in program/remote Owner.
RSP	REAL	Enables the remote speed setpoint, which is usually set by the continuous control, to be connected in program/remote Owner with the selected SP speed (RSpeedSel = 0).
RStart	BOOL	1 = Enables the speed drive to be run in program/remote mode. If RStop is connected then the rising edge (Pulse based control) sends the command to start the speed drive only when program/remote cascade mode is configured.
RStop	BOOL	A rising edge sends a stop command only when program/remote cascade mode is configured.
FWDILCKSpeedSel	UINT	Enables to select which remote setpoint needs to be considered in case of forward direction interlocking. Refer to the possible RSpeedSel values.
FWDILCKPos	REAL	Setpoint needs to be considered when the DFB is FWDILCK = 1 and the analog setpoint is selected (FWDILCKSpeedSel = 0). Refer to the FWDILCKSpeedSel and FWDILCKl inputs.
FWDILCKStart	BOOL	1 = Enables to define the forward direction interlock position on run.
FWDILCK	BOOL	Forward direction interlock is active. Refer to the FWDILCKSpeedSel and FWDILCKPos inputs. **NOTE:** The signal connected to this input will not be effective if the device requires resetting. In that case, it will be de-energized (Forward and Reverse signals to 0)
REVILCKSpeedSel	UINT	Enables to select which remote setpoint needs to be considered in case of reverse direction interlocking. Refer to the possible RSpeedSel values.
REVLCKPos	REAL	Setpoint needs to be considered when the DFB is REVILCK = 1 and the analog setpoint is selected (REVILCKSpeedSel = 0). Refer to the REVILCKSpeedSel and REVILCK inputs.
REVILCKStart	BOOL	1 = Enables to define the reverse direction interlock position on run.

Parameter	Type	Description
REVILCK	BOOL	Reverse direction interlock is active. Refer to the REVILCKSpeedSel and REVLCKPos inputs. **NOTE:** The signal connected to this input will not be effective if the device requires resetting. In that case, it will be de-energized (Forward and Reverse signals to 0)
ExtCTLD	BOOL	1 = Indicates to the DFB the speed drive is being controlled from outside (for example, from the console or from a push-button panel) the system. Enables to avoid the generation of erroneous detections of unsuccessful operation while the device is being controlled from outside the system. Monitoring of the alarm for the detection of unsuccessful operations stops when the device is externally controlled.

MotorVSGP_Engpar_DDT Type

Name	Type	Description
TimeOut	TIME	Allows configuring the time during which the detection of unsuccessful operations is disabled. 0 disables the detection. After the configured time has elapsed, an alarm is generated. Refer to the Alarm output, page 205 and MotorVSGP_ST.STW.Alarm input/output pin, page 208.
ScanTime	TIME	Allows configuring the time during which the detected alarm signals are kept active. Allows the Supervision system to acquire the data for alarms that are automatically reset. Refer to the Alarm and Failed outputs, page 205.
ConfRear-mEn	BOOL	• 1 = Withdraws the output when a non-confirmed operation is detected (*Alarm* = 1) and requires a mandatory manual resetting once the *Alarm* condition disappears. • 0 = The output remains in its previous state when a non-confirmed operation is detected (*Alarm* = 1) and does not need manual resetting, once the *Alarm* condition disappears. Refer to the TimeOut input and the MotorVSGPST.STW.Alarm input/output pin, page 208.
FailRear-mEn	BOOL	• 1 = A manual resetting is required when the detected error/interlock trip alarm condition disappears. • 0 = No manual resetting is required when the detected error/interlock trip alarm condition disappears.
DisStopA-larm	BOOL	1 = Disable device detected error alarm when motor is not running.
ResetSP-Fail	BOOL	Reset setpoint in case of detected error condition (1 = Restart is required if motor trips due to detected error or alarm).
TrackSPEn	BOOL	Track set-point in case of that the interlock condition is active. **NOTE:** The *TrackSPEn* parameter enables the interlock trip alarm when the device stopped due to interlock conditions.

Outputs

Output Parameter Description

Parameter	Type	Description
Forward	BOOL	1= Forward direction field command signal active. It always follow the LForward command. **NOTE:** During simulation the signal will be deactivated.
Reverse	BOOL	1= Reverse direction field command signal active. It always follow the LReverse command. **NOTE:** During simulation the signal will be deactivated.
LForward	BOOL	1 =Forward direction Logical command signal active. The current output (Forward) is continuously copied to this variable, except in the simulation mode. Refer to the LForward and LReverse interlock calculation, page 207.

Parameter	Type	Description
LReverse	BOOL	1 =Reverse direction Logical command signal active. The current output (Reverse) is continuously copied to this variable, except in the simulation mode. Refer to the LForward and LReverse interlock calculation, page 207.
OP	REAL	Speed value corresponding to the current output of the speed drive. The signal (coming from the speed drive) that provides this speed in the engineering units desired for representation in the monitoring subsystem (the DFB does not scale this signal) needs to be connected. It always follow the LOP command. **NOTE:** During simulation the signal will be deactivated.
LOP	REAL	The current output (OP) is continuously copied to this variable, except in the simulation mode.
SPSELD	BOOL	1 = Analog setpoint input activated in the speed drive.
S1SELD S2SELD S3SELD	BOOL	1 = Multispeed *x* activated in the speed drive.
SCode0	BOOL	Multispeed 1 or 3 selected.
SCode1	BOOL	Multispeed 2 or 3 selected.
Running	BOOL	1 = Allows you to determine the speed drive is delivering a frequency to the output. It is evaluated considering the Run and the MotorVSGP_ST.CFGW.Sim, page 208 status. Refer to the Running feedback calculation, page 207.
SP	REAL	1 = Current speed setpoint in the speed drive. Refer to the SP Calculation, page 206.
Alarm	BOOL	1 = Indicates if the detection of unsuccessful operation has occurred. Refer to the Alarm behavior for more details.
Failed	BOOL	1 = Indicates an inoperable device. It is evaluated according to the value of the Fail and if FailRearmEn engineering parameter configured, then Failed will be high until rearm is provided. When it is activated, a time period is set so that the signal is maintained for a minimum of ScanTime. 0 = After the minimum activation time has passed, it is only deactivated if the Fail input signal is inactive.
Normal	BOOL	1 = Indicates the motor is operating in normal mode.
OOS	BOOL	1 = Indicates the equipment is out of service.
NReady		1 = Indicates motor is not ready for operation.

The table shows the value of the coding based on the status of the SPSELD, S1SELD, S2SELD, and S3SELD outputs:

Selected multispeed	SCODE0	SCODE1
SP (SPSELD = ON)	OFF	OFF
S1 (S1SELD = ON)	ON	OFF
S2 (S2SELD = ON)	OFF	ON
S3 (S3SELD = ON)	ON	ON

SP Calculation

The signal evaluation depending on the SC.Rem and RSP inputs and the MotorVSGP_ST.CFGW.Owner input/output is shown in the following table:

Owner	Rem	SP is calculated as
OFF	OFF	Speed selected (SC.LSP, SC.SELSP, SC.SELS1, SC.SELS2, SC.SELS3 public variables and the S1, S2, and S3 inputs) from the sequential control.
OFF	ON	Speed selected from the continuous control (RSP, RSpeedSel, S1, S2, and S3 inputs).
ON	-	Speed selected from the continuous control (MotorVSGP_ST.SP and MotorVSGP_CFG.SPSpeedSel inputs/outputs and the S1, S2, and S3 inputs.

Interlock Calculation

LForward and LReverse in interlock state:

When LForward = OFF and LReverse = ON					
FWDILCK	FWDILCK-Start	REVILCK	RE-VLCKSt-art	LForward	LReverse
ON	OFF	OFF	OFF	Holds the last output state (Operator can change setpoint to Stop).	
ON	ON	OFF	OFF	ON	OFF
ON	OFF	ON	OFF	OFF	OFF
ON	OFF	ON	ON	OFF	ON
ON	ON	OFF	ON	ON	OFF
ON	ON	ON	ON	Holds the last output state.	
OFF	OFF	ON	OFF	OFF	OFF
OFF	OFF	ON	ON	OFF	ON
OFF	-	OFF	-	Holds the last output state.	

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

The *ConfRearmEn* engineering parameter has to be set to 1 for the *OP* pin of the function block output to become 0, under non-confirmed operations.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: Non-confirmed operations are those for which commands have been given and to which there is no confirming reply from the equipment.

Feedback Calculation for Running

RunFB connected	Simulation	Override	Running is calculated as:
-	-	ON	LForward or LReverse
-	ON	OFF	LForward or LReverse and (TimeOut/2)
OFF	OFF	OFF	LForward or LReverse
ON	OFF	OFF	RunFB

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
MotorVSGP_CFG	MotorVSGP_CFG_DDT	Data structure of the DFB configuration.
MotorVSGP_ST	MotorVSGP_ST_DDT	Data structure of the DFB status. Usually used from the monitoring subsystem for monitoring the status of the speed drive.

MotorVSGP_CFG_DDT Type

Name	Type	Description
SPSpeed-Sel	UINT	Enables to select which setpoint needs to be used if the owner is the operator. This data is write-only if the owner is the operator (OWNER is 1), and is loaded with the selected value if the owner is the program (OWNER is 0). Refer to the possible RSpeedSel, page 203 values.
Reserved	UINT	Reserved
S1	REAL	Read-only access Refer to the S1 input pin, page 203.
S2	REAL	Read-only access Refer to the S2 input pin, page 203.
S3	REAL	Read-only access Refer to the S3 input pin, page 203.

MotorVSGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
PV	–	–	REAL	Refer to PV input pin, page 203.
SP	–	–	REAL	Refer to SP output pin, page 205.
LOP	–	–	REAL	Refer to LOP output pin, page 205.
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	Running	*Bit0*	BOOL	Refer to Running output pin, page 205.
	LForward	*Bit1*	BOOL	Refer to LForward output pin, page 205.
	LReverse	*Bit2*	BOOL	Refer to LForward output pin, page 205.
	ReverseEn	*Bit3*	BOOL	Refer to ReverseEn input pin, page 203.
	Alarm	*Bit4*	BOOL	Refer to Alarm output pin, page 205.
	Failed	*Bit5*	BOOL	Refer to Failed output pin, page 205.
	Rem	*Bit6*	BOOL	Refer to Rem.
	ILCK	*Bit7*	BOOL	1 = Interlock condition is active (FWDILCK or REVILCK). **NOTE:** If the device requires resetting (STW.Reamr = 1), the OP takes the value = 0 regardless of the ILCKPos value waiting for the operator to carry out the reset.
	RearmR	*Bit8*	BOOL	Refer to RearmR.
	ExtCTLD	*Bit9*	BOOL	Read-only access

Name	Bit Name	Bit	Type	Description
				Refer to ExtCTLD input pin, page 203.
	SPSELD	*Bit10*	BOOL	Read-only access Refer to SPSELD output pin, page 205.
	S1SELD	*Bit11*	BOOL	Read-only access Refer to S1SELD output pin, page 205.
	S2SELD	*Bit12*	BOOL	Read-only access Refer to S2SELD output pin, page 205.
	S3SELD	*Bit13*	BOOL	Read-only access Refer to S3SELD output pin, page 205.
	SPEn	*Bit14*	BOOL	Read-only access Refer to SPEn input pin, page 203.
	PowerRMVL	*Bit15*	BOOL	Read-only access Refer to PowerRMVL input pin, page 203.
STW1	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	Nready	*Bit0*	BOOL	Refer to Nready output pin, page 205.
	FWDSPLocked	*Bit1*	BOOL	1 = Forward direction setpoint lock active.
	REVSPLocked	*Bit2*	BOOL	1 = Reverse direction setpoint lock active.
	FWDILCKTrip	*Bit3*	BOOL	1 = Forward direction interlock trip active.
	REVILCKTrip	*Bit4*	BOOL	1 = Reverse direction interlock trip active.
	FWDILCK	*Bit5*	BOOL	Refer to FWDILCK input pin, page 203.
	REVILCK	*Bit6*	BOOL	Refer to REVILCK input pin, page 203.
	Fail	*Bit7*	BOOL	Refer to Fail input pin, page 203.
	FailAlarm	*Bit11*	BOOL	1 = Detected error alarm is active.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read-only access to the data contained in this word.
	Owner	*Bit0*	BOOL	Read/write access Refer to Owner, page 210.
	ILCKBP	*Bit1*	BOOL	Read/write access 1 = Enables the interlock to be bypassed.
	Override	*Bit2*	BOOL	Refer to Override, page 210.
	Rearm	*Bit3*	BOOL	Write access Refer to Rearm, page 210.
	LSPRev	*Bit4*	BOOL	Read/write access Refer to LSPRev, page 210.
	LStart	*Bit5*	BOOL	Read/write access Refer to LStart, page 210.
	SimFBFail	*Bit12*	BOOL	1 = Feedback will not simulate in simulation mode.
	Sim	*Bit13*	BOOL	Read/write access Refer to Sim, page 210.
	Maint	*Bit14*	BOOL	Refer to Maint, page 210.
	OOS	*Bit15*	BOOL	Read/write access Refer to OOS output pin, page 205.

Public Variables

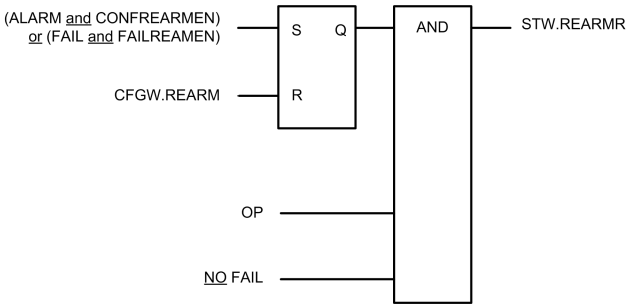
Public Variable Description

Variable	Type	Description
SC	MotorVSGP_SC_DDT	Provides the frequently needed data to monitor and control the regulator status from the sequential control.

MotorVSGP_SC_DDT Type

Name	Type	Description
PV	REAL	Read-only access Refer to PV input pin, page 203.
SP	REAL	Read-only access Refer to SP output pin, page 205.
OP	REAL	Read-only access Refer to OP output pin, page 205.
LOP	REAL	Read-only access Refer to LOP output pin, page 205.
Rem	BOOL	Read/write access • 1 = Enables the DFB to be configured to remote set-point Rem. • 0 = Enables the DFB to be configured to local set-point LSP.
LSPREV	BOOL	Read/write access 1 = Enables to set the reverse rotation direction (0: forward, 1: reverse) in Program/Local Owner. 0 = Enables to set the forward rotation direction in Program/Local Owner. In other Owner, the DFB continuously assigns the current rotation direction set-point to it.
LSTART	BOOL	Read/write access 1 = Enables the speed driver to run in Program/Local Owner. Read/write access Enables to set the applicable speed set-point in Program/Local mode when the speed has been selected through the SC.SELSP variable. The DFB sets this signal to 0 after the signal has been processed.
LSP	REAL	Read/write access Enables the assignment of the local set-point for the sequential control if the owner is the Program (Owner is 0) and the selected set-point is Local (Rem is 0). In any other case, the DFB automatically assigns it the current value of the resulting set-point to it (refer to the SP output pin, page 205).
LStart1	BOOL	Read/write access 1 = Rising edge to enable Forward direction Start command while the owner is the Program and Local.
LStart2	BOOL	Read/write access 1 = Rising edge to enable Reverse direction Start command while the owner is the Program and Local.
LStop	BOOL	Read/write access 1 = Rising edge to enable Stop command while the owner is the Program and Local.
Forward	BOOL	Read-only access Refer to Forward output pin, page 205.
Reverse	BOOL	Read-only access Refer to Reverse output pin, page 205.
LForward	BOOL	Read-only access

Name	Type	Description
		Refer to LForward output pin, page 205.
LReverse	BOOL	Read-only access Refer to LReverse output pin, page 205.
Running	BOOL	Read-only access Refer to Running output pin, page 205.
Alarm	BOOL	Read-only access Refer to Alarm output pin, page 205.
Failed	BOOL	Read-only access Refer to Failed output pin, page 205.
FailAlarm	BOOL	Read-only access 1 = Detected error alarm is active.
ILCKD	BOOL	Read-only access The signal evaluation depending on the ILCK input and the MotorVSGP_ST.CFGW.ILCKBP input/output is shown as follows: **ILCKD** / **ILCKBP** / **SC.ILCKD** OFF / OFF / OFF OFF / ON / OFF
FWDILCKD	BOOL	Read-only access Forward direction Interlock and not bypassed.
REVILCKD	BOOL	Read-only access Reverse direction Interlock and not bypassed.
SPSELD	BOOL	Read-only access Refer to SPSELD output pin, page 205.
S1SELD	BOOL	Read-only access Refer to S1SELD output pin, page 205.
S2SELD	BOOL	Read-only access Refer to S2SELD output pin, page 205.
S3SELD	BOOL	Read-only access Refer to S3SELD output pin, page 205.
PowerRMVL	BOOL	Refer to PowerRMVL input pin, page 203.
SELSP	BOOL	Read/write access Enables the speed set-point to be set in Program/Local mode to the SC.LSP speed. 1 allows this speed to be selected. The DFB sets this signal to 0 after the signal has been processed.
SELS1 SELS2 SELS3	BOOL	Read/write access 1 = Enables the speed set-point to be set in Program/Local mode to the speed driver multispeed x. 1 enables this speed to be selected. The DFB sets this signal to 0 after the signal has been processed. **NOTE:** If more than one speed is selected simultaneously, the following order of priority applies: SP, S1, S2, and S3.
Owner	BOOL	Read/write access Enables to configure whether the set-point is set by the Program (0) or by the Operator (1).
Rearm	BOOL	Read/write access Allows the device to be reset (1). The block sets the signal to 0 after each execution and sets the signal indicating that the resetting is required (CFGW.Rearm) to 0.
RearmR	BOOL	Read/write access Indicates if the device requires resetting (1) after the detection of an inoperable device condition if it has been configured through the ConfRearmEn and/or FailRearmEn inputs.

Name	Type	Description
		The signal activation logic included in the DFB is as follows. 
`Normal`	BOOL	Read/write access Refer to `Normal` output pin, page 205.
`Override`	BOOL	Read/write access Enables to set the device to `Override` mode so that the actual position signals (`Run` input) are ignored. It is simulated that the device is in the desired position (according to the `OP` output pin) and is functional.
`Sim`	BOOL	Read/write access Enables to set the device to simulation mode so that the actual position signals (`Run` input) are ignored. It is simulated that the device is in the desired position after the half time of timeout value (according to the `LOP` output pin) and suppressed the `OP` output pin. During running or transition state it is not possible to change simulation mode.
`Maint`	BOOL	Read-only access 1 = Maintenance mode active. Allows the operation of the motor for maintaining and testing. Interlock conditions are bypassed and detected error conditions do not impact the output. During running or transition state it is not possible to change maintenance mode.
`OOS`	BOOL	Read/write access Refer to `OOS` output pin, page 205.
`NReady`	BOOL	Read-only access Refer to `NReady` output pin, page 205.
`ExtCTLD`	BOOL	Refer to `ExtCTLD` input pin, page 203.

Optional Service - Local Panel (MOTORVSLPGP)

Overview

This chapter describes the optional services - local panel (MOTORVSLPGP) DFB.

Description

General

The objective of the `MOTORVSLPGP` DFB is to manage a local panel that controls a motorized valve. This valve is implemented with an `MOTORVSGP` DFB and with signals that are wired to the controller so that the latter defines the target position for the device.

Function Description

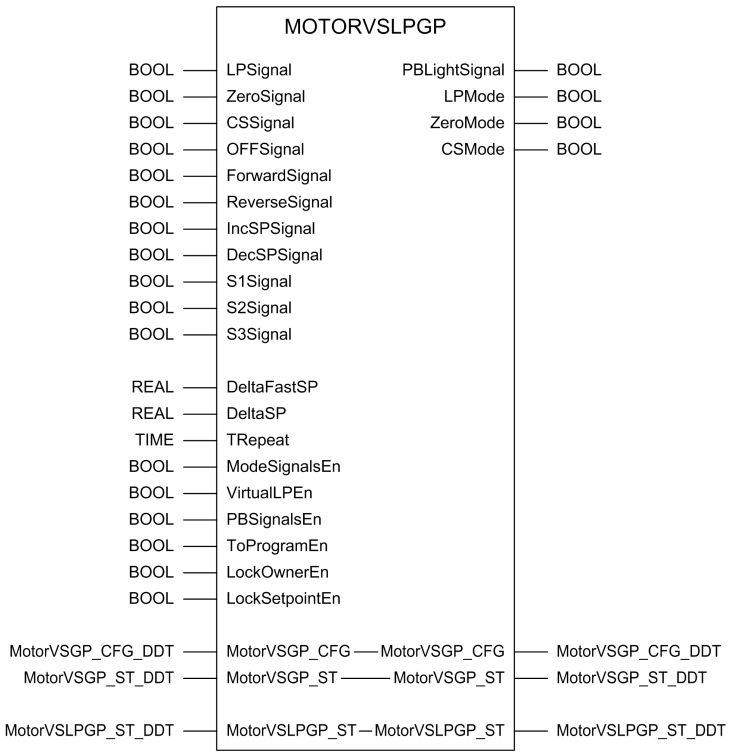
The main functions of the module are described in the following table:

Function	Description
Mode Switch	Optionally manages the signals coming from an operating mode switch in a local panel with the following configuration: Local - Zero - Control System. The Zero mode signal is optional. However the user can enable Local/ Control System mode from the faceplate also, when *VirtualLPEn* input pin signal is high and *ModeSignalsEn* signal is low in the DFB.
Push buttons	The DFB manages up to two signals coming from OFF and ON push buttons of local panel, giving the OFF push button higher priority.
Owner Management	The DFB enables to configure whether the Program needs to be the device owner or not, after switching to the Control System mode again.
Owner Locking	The DFB enables to configure whether the monitoring (HMI) system needs to block access to the drop-down list or not, it disables the access to change the owner (Operator/Program) while the device is controlled from the Local Panel.
Set-point Lock	The DFB enables to configure whether the monitoring (HMI) system needs to block access to the drop-down list or not. It disables to change the set-point while the device is controlled from the Local Panel.
Push button Enabling/Disabling	The push button signals from the Local Panel can be enabled/disabled through the DFB configuration (input pin) and/or from control sequences.
Enabled Panel Signaling	The DFB provides a signal that can be used to illuminate a light source on the Local Panel to indicate when the push buttons are enabled for operation.
Virtual Panel Enabling/Disabling	This signal enables the operator to select the Local/Control System mode of operation from the HMI so that the push button signals are enabled for operation.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Name	Type	Description
LPSignal	BOOL	1 = The local panel mode switch is in the Local Panel position. Refer to the table included with the description of the LPMode output pin, page 215.
ZeroSignal	BOOL	1 = The local panel mode switch is in the zero position.This signal is optional even if a mode switch is available on the local panel. When this signal is a logical high, user is not allowed to operate the device from the faceplate and also from the DFB. Hence, all the functions in the faceplate will be disabled. Refer to the table included with the description of the LPMode output pin, page 215.
CSSignal	BOOL	1 = The local panel mode switch is in the Control System position. Refer to the table included with the description of the LPMode output pin, page 215.
OFFSignal	BOOL	1 = The OFF push button on the local panel is pressed. **NOTE:** OFF push button signal has higher priority than ON push button signal.
ForwardSignal	BOOL	Forward push button pressed.
ReverseSignal	BOOL	Reverse push button pressed.
IncSPSignal	BOOL	Increase setpoint in DeltaSP, if SPEn = 1.
DecSPSignal	BOOL	Decrease setpoint in DeltaSP, if SPEn = 1.
S1Signal	BOOL	Select S1 speed.
S2Signal	BOOL	Select S2 speed.
S3Signal	BOOL	Select S3 speed.
DeltaFastSP	BOOL	Absolute set-point increase or decrease value when the Open and Close push buttons are pressed.
DeltaSP	BOOL	Absolute set-point increase or decrease value when the Open and Close push button is pressed.
TRepeat	BOOL	Indicates the maximum time that elapses before the system interprets that a command has been repeated when the push button is pressed.
ModeSignalsEn	BOOL	1 = Enables the use of the Local Panel/Zero/Control System (or Local Panel - Control System) mode switch on the local panel. Usually configured in development phase based on the characteristics of the local panel being used. If the mode switch is enabled, the LPSignal, ZeroSignal, and CSSignal inputs are considered to determine the operating modes of the local panel. Refer to the table included with the description of the LPMode output pin, page 215. **NOTE:** *ModeSignalsEn* input pin has higher priority than the *VirtualLPEn* input pin signal.
VirtualLPEn	BOOL	1 = This signal enables the operator to select the Local/Control System mode of operation from the HMI, so that the push button signals are enabled for operation. **NOTE:** When the owner is program, the Local/Control System mode selection drop-down list will be visible but disabled for operation however, when the owner is operator the Local Panel drop-down list will be accessible for operation. Refer to the table included with the description of the LPMode output pin, page 215.
PBSignalsEn	BOOL	1 = Enables to accept the OFF and ON push button signals from the local panel. This signal is not applicable if the mode switch is enabled (refer to the ModeSignalsEn input). It means that push button signals are considered when in Local Panel mode. When the mode switch is disabled (ModeSignalsEn = 0), the PBSignalsEn signal enables/disables the push button signals from the control system itself based on the relevant process conditions. The SC.DisableLP public variable is also considered for determining whether the push buttons are enabled or not when *LPMode* pin signal is high, as shown in the following table.

VirtualLPEn	ModeSignalsEn	PBSignal-sEn	SC.DisableLP	Push buttons enabled?
OFF	OFF	OFF	-	NO
OFF	OFF	-	ON	NO
OFF	OFF	ON	OFF	YES
OFF	ON	-	-	YES
ON	OFF	OFF	-	YES

Therefore, this signal can be used to enable/disable the push buttons on a simple local panel that does not feature a mode switch (Local Panel/Zero/Control System) based on relevant process conditions.

Name	Type	Description
ToProgramEn	BOOL	1 = Change to program mode when control is switched to control system mode and VirtualLPEn = 0.
LockOwnerEn	BOOL	1 = Disables the access of changing the owner from HMI while the mode switch is in local panel. The following table shows how the on-off device owner is evaluated based on *ToProgramEn*, *LockOwnerEn* input signals and the local panel operating mode:

Local Panel Mode	ToProgramEn	LockOwnerEn	Owner
Zero mode	-	-	Operator
Switch to Local Panel	-	-	Operator
Local Panel	-	ON	Operator
Switch to Control System	ON	-	Program

In remaining cases, owner of the on/off device ceases to be defined from the MotorLPGP DFB and it enables to change it from monitoring (HMI) system with the command word in MotorLPGP_ST.CFGW input/output of the corresponding MotorLPGP DFB.

In any case, consider that, the device owner that had been set before switching to the local panel mode is not memorized.

Name	Type	Description
LockSetpointEn	BOOL	1 = Disables the access to set point of on/off devices in the monitoring (HMI) system when it is in local panel mode only, and the owner is operator. Therefore, user can activate this signal to lock setpoint operation from the monitoring (HMI) system while on/off device is controlled from the local panel.

Outputs

Output Parameter Description

Name	Type	Description
PBLight-Signal	BOOL	1 = The push buttons on the local panel are fully operational, that is, the local panel is in Local Panel mode (either because the mode is selected with the corresponding switch or because the push buttons are enabled in the event that there is no selector switch available or the *VirtualLPEn* is high with the local panel mode enabled).
LPMode	BOOL	1 = The local panel mode is in Local Panel. The following table shows how the local panel operating mode is determined based on the ModeSignalsEn, VirtualLPEn, PBSignalsEn, LPSignal, CSSignal and ZeroSignal, input signals:

ModeSignalsEn	VirtualLPEn	PBSignalsEn	LPSignal	CSSignal	ZeroSignal	Output
0	0	0	0	0	0	CS mode
1	0	0	0	0	0	Zero mode
1	0	0	1	0	0	LP mode and PBLight-Signal
1	0	0	0	1	0	CS mode
0	1	1	NA	NA	NA	CS mode
0	1	0	NA	NA	NA	CS mode
0	0	1	0	0	0	LP mode and PBLight-Signal
1	1	0	1	0	0	LP mode and PBLight-Signal

Name	Type	Description						
		1	1	0	0	0	0	Zero mode
		1	1	0	0	0	1	Zero mode
		1	1	0	0	1	0	CS mode
		1	0	1	0	1	0	CS mode
		1	1	1	0	1	0	CS mode
		1	0	0	0	0	1	Zero mode
ZeroMode	BOOL	1 = The local panel mode is in Zero. User is not allowed to operate the device from the faceplate and also from DFB. Hence, all the functions in the faceplate will be disabled. Refer to the table included with the description of the LPMode output.						
CSMode	BOOL	1 = The local panel mode is in Control System. Refer to the table included with the description of the LPMode output.						

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
MotorVSGP_CFG	MotorVSGP_CFG_DDT, page 208	Data structure corresponding to the motorized valve is to be controlled from the local panel (for detailed information regarding this structure, refer the MOTORVSGP DFB). The structure provides the information needed for the operation of the MOTORVSLPGP DFB. It enables this DFB to control the set-point for the motorized valve.
MotorVSGP_ST	MotorVSGP_ST_DDT, page 208	Data structure corresponding to the motorized valve is to be controlled from the local panel (for detailed information regarding this structure, refer to the MOTORVSGP DFB, page 203.
MotorVSLPGP_ST	MotorVSLPGP_ST_DDT	Data structure that is used as an interface with the monitoring (HMI) system. The structure provides the information needed for the operation of the MOTORVSLPGP DFB and enables this DFB to control the owner of the motorized valve.

MotorVSLPGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	LPMode	*Bit0*	BOOL	Refer to the LPMode output pin, page 215.
	ZeroMode	*Bit1*	BOOL	Refer to the ZeroMode output pin, page 215.
	CSMode	*Bit2*	BOOL	Refer to the CSMode output pin, page 215.
	OwnerLocked	*Bit3*	BOOL	Indicates whether the owner change buttons in the monitoring (HMI) system needs to be locked (1) or not (0). Is activated when: • The Local Panel mode is set to Zero or • It is set to Local Panel and the LockOwnerEn input signal has been configured as active (1).
	SPLocked	*Bit4*	BOOL	Indicates whether the set-point change and START buttons in the monitoring (HMI) system needs to be locked (1) or not (0). Is activated when: • The Local Panel mode is set to Zero or

Name	Bit Name	Bit	Type	Description
				• It is set to Local Panel and the `LockSetpointEn` input signal has been configured as active (1).
	`ModeSignalsEn`	*Bit5*	`BOOL`	Refer to the `ModeSignalsEn` input pin, page 214.
	`PBEnabled`	*Bit6*	`BOOL`	Refer to the `PBEnabled` input pin, page 214.
	`VirtualLPEn`	*Bit7*	`BOOL`	Refer to the `VirtualLPEn` input pin, page 214.
	`LPSignal`	*Bit8*	`BOOL`	Refer to the `LPSignal` input pin, page 214.
	`ZeroSignal`	*Bit9*	`BOOL`	Refer to the `ZeroSignal` input pin, page 214.
	`CSSignal`	*Bit10*	`BOOL`	Refer to the `CSSignal` input pin, page 214.
`STW`	`OffSignal`	*Bit11*	`BOOL`	Refer to the `OffSignal` input pin, page 214.
	`ForwardSignal`	*Bit12*	`BOOL`	Refer to the `ForwardSignal` input pin, page 214.
	`ReverseSignal`	*Bit13*	`BOOL`	Refer to the `ReverseSignal` input pin, page 214.
	`IncSPSignal`	*Bit14*	`BOOL`	Refer to the `IncSPSignal` input pin, page 214.
	`DecSPSignal`	*Bit15*	`BOOL`	Refer to the `DecSPSignal` input pin, page 214.
`CFGW`	–	–	`WORD`	Provides the means to control the device from the monitoring subsystem. Read-only access to the data contained in this word.
	LPEnabled	*Bit0*	`BOOL`	Read/write access • 1 = Enables the Local Panel mode. • 0 = Enables the Control System mode.

Public Variables

Public Variable Description

Variable	Type	Description
`SC`	`MotorVSLPGP_SC_DDT`	Provides the frequently needed data for monitoring and controlling the DFB.

`MotorVSLPGP_SC_DDT` Type

Name	Type	Description
`LPMode`	`BOOL`	Read-only access Refer to the `LPMode` output pin, page 215.
`ZeroMode`	`BOOL`	Read-only access Refer to the `ZeroMode` output pin, page 215.
`CSMode`	`BOOL`	Read-only access Refer to the `CSMode` output pin, page 215.
`DisableLP`	`BOOL`	Read/write access 1 = Disables the push buttons on the local panel. Only applicable when a local panel without mode switch configuration is used (`ModeSignalsEn` = 0).

Supervision

Overview

This section describes the Supervision functions and runtime services that are available for the variable speed motor.

Genies

Genie Properties

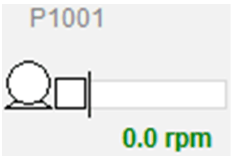
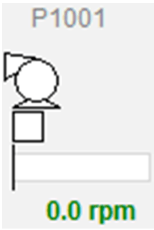
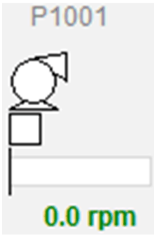
Refer to Genie Properties, page 62

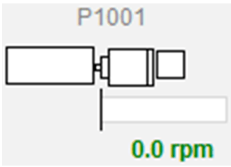
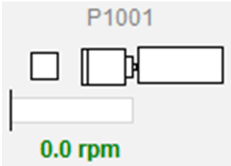
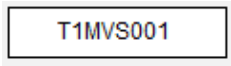
Genie Icons

Refer to Genie Icons, page 63

Representation

The table describes the genies of the `gpl_motorvs` library:

Graphic symbol	Genie name	Description
	Motor_Vs	Variable speed motor.
	Motor_Vs_Up	Variable speed motor and up directions.
	Pump_Vs_Left	Variable speed motor and centrifugal pump left with Current value (`PV`) and set-point (`SP`).
	Pump_Vs_Right	Variable speed motor and centrifugal pump right with Current value (`PV`) and set-point (`SP`).

Graphic symbol	Genie name	Description
	Screwpump_Vs_Left	Variable speed motor and screw pump left with Current value (`PV`) and set-point (`SP`).
	screwpump_Vs_Right_	Variable speed motor and screw pump left with Current value (`PV`) and set-point (`SP`).
	motorvsinstanceid_m	Variable speed motor instance.

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

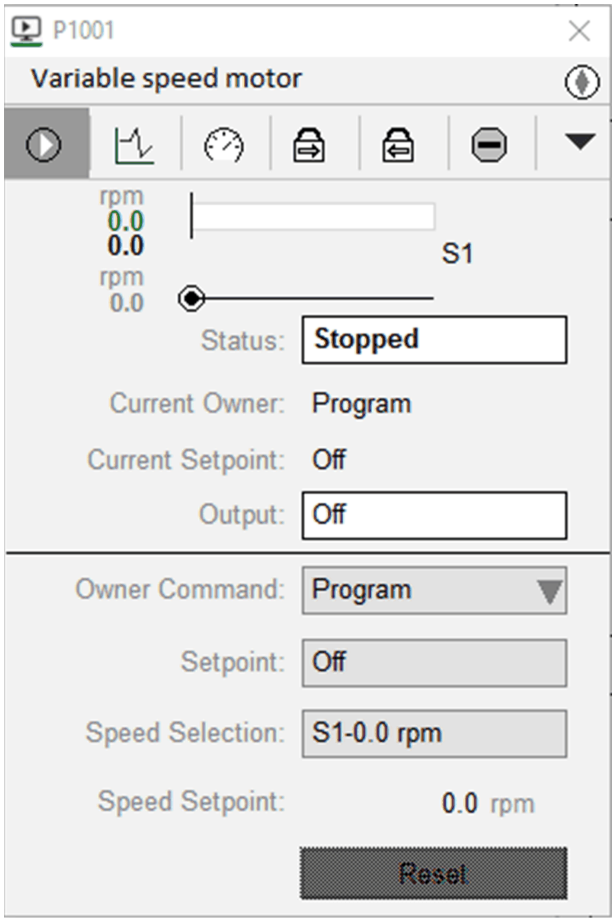
Available Tabs

During operation, clicking an motorVs genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
  - Event, page 69
- Tabs for optional functions, which appear only if configured:
  - Interlock, page 87
  - Failure, page 78
  - Measures, page 70

Operator Tab

The figure shows an operator tab:

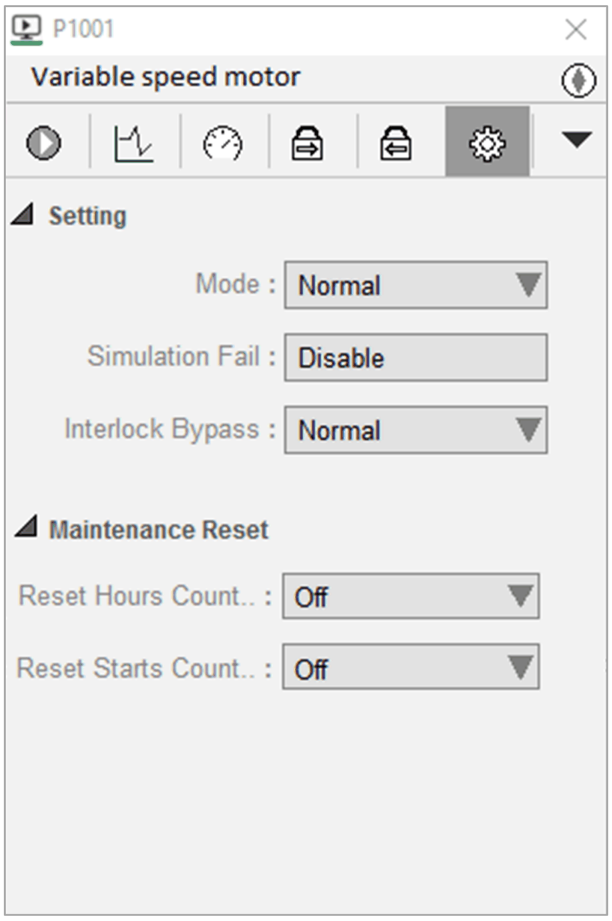


The table describes the control function of an operator tab:

Label	Item name	Description	Security parameter, page 200
Status	EquipStatus	Equipment status of motorVs. For more details, refer toEquipStatus, page 159.	–
Current Owner	Owner	Current owner of motorVs. For more details, refer to Owner, page 58.	
Current Setpoint	CurrentSP	Current setpoint of motorVs.	
Output	EquipState	Output status of motorVs. For more details, refer to EquipState, page 159.	
OwnerCommand	OwnerCommand	Owner command of motorVs (Program, Operator or Local Panel). **NOTE:** Local panel owner is enabled when the virtual panel is selected.	Set owner
Setpoint	OperatorSetpoint	Allows you to change the setpoint, if owner = operator.	Set setpoint
Speed Selection	SpeedSetpointSelec-tion	Allows you to select the motor speed.	Set mode override
Speed setpoint	Setpoint	Current setpoint of motorVs in RPM.	–
–	Rearm	Allows you to reset manually after the detection of an inoperable device condition or unsuccessful operation. **NOTE:** When the control module is reset, the current setpoint shown in operator tab is effective. Also, when you click **Reset**, a dialog box opens to confirm.	Set reset

Engineering Tab

The figure shows an engineering tab:



The table describes the control function of engineering tab:

Label	Item name	Description	Security parameter, page 200
Setting			
Mode	Mode	Allows you to view and change the mode. For more details, refer to operating mode, page 59.	Set mode - normal, set mode - override, set mode - simulation, set mode - maintenance, and Set mode - out of service **NOTE:** Each configurable mode has the different security parameters
Simulation Fail	SimulationFailActive	Enable or disable feedback simulation. **NOTE:** The Simulation Fail will only be available when the parameter `SimulationFail` is created in the citect parameters. To create, go to **Setup** > **Parameters**. Enter the `GPL.Mode` in **Section Name**, `SimulationFail` in **Name** and set the **Value** = 1.	Set mode - simulation
Interlock Bypass	InterlockBypass	Interlock bypass status Allows you to bypass interlock.	Set bypass conditions, set bypass all interlocks
Maintenance Reset			
Reset Hours Count	MotMntResetHour-sCounter	Reset the hours counter.	Set reset
Reset Starts Count	MotMntResetManeuver-sCounter	Reset the operations counter.	Set reset

Items

Overview

This section describes the variables, equipstatus and equipstate variable, abnormal and action required conditions, alarms, events, and trends items of `variable speed motor`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`FeedbackRunning`	Run feedback	Running/Not Running	*Bit0*
`OutputForward`	Forward feedback	–	*Bit1*
`OutputReverse`	Reverse feedback		*Bit2*
`ReverseEnable`	Reverse direction	Enabled/Disabled	*Bit3*
`AlarmActive`	Feedback confirmation detected error	Active/Inactive	*Bit4*
`FailActive`	Motor detected error	Active/Inactive	*Bit5*
`ProgramSetpointMode`	Setpoint source for program owner	Remote/Sequence	*Bit6*
`InterlockActive`	Interlock status	Interlocked/Healthy	*Bit7*
`RequireRearm`	Reset required	Required/Not Required	*Bit8*
`OwnerExternal`	External owner status	External/Not External	*Bit9*
`SelectedSpeedSet-point`	Analog setpoint enable	Selected/Not Selected	*Bit10*
`SelectedSpeed1`	Speed 1 enable	Enabled/Disabled	*Bit11*
`SelectedSpeed2`	Speed 2 enable		*Bit12*
`SelectedSpeed3`	Speed 3 enable		*Bit13*
`SetpointEnable`	Motor setpoint enable status		*Bit14*
`PowerRemoval`	Speed drive disabled	On/Off	*Bit15*
Bit items derived from StatusWord1.			
`NotReady`	Not ready to function	–	*Bit0*
`Setpoint1LockActive`	Forward direction setpoint change is denied		*Bit1*
`Setpoint2LockActive`	Backward direction setpoint change is denied		*Bit2*
`Interlock1Trip`	Forward direction interlocked trip is active		*Bit3*
`Interlock2Trip`	Backward direction interlocked trip is active		*Bit4*
`Interlock1Active`	Interlocked trip is active		*Bit5*

Item name	Description	Enumeration	Address
Interlock2Active	Interlocked trip is active		*Bit6*
FailAlarmActive	Detected error alarm is active		*Bit11*
Bit items derived from ConfigurationWord.			
OwnerSelect	Owner of control module	Operator/Program	*Bit0*
InterlockBypass	Interlock bypass command	Bypassed/Normal	*Bit1*
InputSource	Input source selection	Operator/Field	*Bit2*
Rearm	Rearm motor	Active/Inactive	*Bit3*
DirectionSetpoint	Operator direction command	Reverse/Forward	*Bit4*
Start	Motor start command	Start/Stop	*Bit5*
SimulationFailAc-tive	Feedback detected error in simulation mode	Enable/Disable	*Bit12*
Simulation	Simulation mode	–	*Bit13*
Maintenance	Maintenance mode		*Bit14*
OutOfService	Motor unavailable	–	*Bit15*
IO Items			
StatusWord	Status word	–	MOTORVSGP_ST.STW
StatusWord1	Status word 1	–	MOTORVSGP_ST.STW1
ConfigurationWord	Configuration word		MOTORVSGP_ST.CFGW
Calculated Items			
Equipstatus	Current status of the variable speed motor. For more details, refer to EquipStatus calculated variable, page 225.	–	GPL_MotorVSStatus
EquipState	Output information For more details, refer to EquipState calculated variable, page 225.		GPL_ MotorVSEquipState
Owner	Current owner of the variable speed motor. For more details, refer to calculated variable, page 58.		GPL_ OwnerExternalLo-cal
CurrentSP	Current setpoint of the variable speed motor.		GPL_ MotorVSCurrent-Setpoint
OperatorSetpoint	Operator setpoint of the variable speed motor.		GPL_ MotorVSOperator-Setpoint
SpeedSetpointSelec-tion	Speed setpoint selection of the variable speed motor.		GPL_ MotorVSDisplay-SpeedSetpoint
SpeedSetpoint	Speed setpoint of the variable speed motor.		GPL_ MotorVSSpeedSet-point

Item name	Description	Enumeration	Address
Mode	Current mode of the variable speed motor. For more details, refer to calculated variable, page 59.		GPL_Mode
Abnormal	Shows abnormal condition of the variable speed motor. For more details, refer to abnormal conditions, page 225.		GPL_Abnormal
ActionRequired	Shows action required condition of the variable speed motor. For more details, refer to action required conditions, page 226.		GPL_ ActionRequired
Optional Items			
Local panel			
Bit items derived from LPStatusWord.			
LocalPanelMode	Control from field	Enabled/Disabled	*Bit0*
ZeroMode	variable speed motor control disabled		*Bit1*
ControlSystemMode	Control from remote		*Bit2*
LockedOwner	Owner change disabled from faceplate	Locked/Unlocked	*Bit3*
LockedSetpoint	Setpoint change disabled from faceplate		*Bit4*
LocalPanelEnabled	Control from field	Enabled/Disabled	*Bit5*
PushButtonEnabled	Local panel push buttons		*Bit6*
VirtualLocalPanelEnabled	Local panel enable from control		*Bit7*
LocalPanelSelected	Control from field	Selected/Not Selected	*Bit8*
ZeroSelected	Control disabled		*Bit9*
ControlSystemSelected	Control from remote		*Bit10*
PushButtonStop	Stop push button signal	On/Off	*Bit11*
PushButtonForward	Forward push button signal		*Bit12*
PushButtonReverse	Reverse push button signal		*Bit13*
PushButtonIncrease	Stop push button signal		*Bit14*
PushButtonDecrease	Start push button signal		*Bit15*
Bit items derived from LPConfigurationWord.			
VirtualSwitch	Local panel selection from faceplate	Local/Control System	*Bit0*

Item name	Description	Enumeration	Address
IO Items			
LPStatusWord	Local panel status word	–	MOTORVSLPGP_ST.STW
LPConfigurationWord	Local panel configuration word		MOTORVSLPGP_ST.CFGW
Failures	Refer to the topic describing the variable items of failures, page 61.		
Interlocks	Refer to the topic describing the variable items of interlocks, page 60.		
Maintenance	Refer to the topic describing the variable items of maintenance, page 59.		

EquipStatus Calculated Variable

The table describes the *EquipStatus* calculated variable:

Equipment Status	Conditions	Enumeration
–	FwdCmd x Run x RevCmd	Unknown
MotorVS Transition	Not FwdCmd x Not RevCmd x Run	Stopping
MotorVS Stopped	Not Run x Not FwdCmd x Not RevCmd	Stopped
MotorVS Forward	Run x FwdCmd x (Not Run + Not RevCmd)	ForwardRunning
MotorVS Reverse	(Not Run + Not FwdCmd) x Run x RevCmd	ReverseRunning
MotorVS Transition	FwdCmd x Not Run	StartingForward
	RevCmd x Not Run	StartingReverse

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	Conditions	Enumeration
Passive with Out of Service	OOS	Off
Passive	Not OOS x Fail x Not Maint	Failed
Active Forward with Interlock	Not OOS x Not Fail x ILCK1 x Not Maint x FwdCmd	Forward Interlocked
Active Reverse with Interlock	Not OOS x Not Fail x ILCK2 x Not Maint x RevCmd	Reverse Interlocked
Passive with Interlock	Not OOS x Not Fail x Not FwdCmd x Not RevCmd	Interlocked
Active Forward	Not OOS x Not Fail x Not ILCK x FwdCmd x Maint	Forward
Passive to Active	Not OOS x Not Fail x Not ILCK x FwdCmd x Not Maint x Not Run	Forward
Active	Not OOS x Not Fail x Not ILCK x FwdCmd x Not Maint x Run	Forward
Active Reverse	Not OOS x Not Fail x Not ILCK x RevCmd x Maint	Reverse
Passive to Active	Not OOS x Not Fail x Not ILCK x RevCmd x Not Maint x Not Run	Reverse
Active	Not OOS x Not Fail x Not ILCK x RevCmd x Not Maint x Run	Reverse
Passive with Out of Service	Not OOS x Not Fail x Not ILCK x Not RevCmd x Maint	Off
Passive	Not OOS x Not Fail x Not ILCK x Not RevCmd x Not Maint x Run	Off
Passive	Not OOS x Not Fail x Not ILCK x Not RevCmd x Not Maint x Not Run	Off

Abnormal Conditions

The table describes the abnormal conditions of variable speed motor:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
`AbnormalOwner`	Abnormal owner status appears if the current owner is not a normal owner which configured at the instance level.	*Bit0*
`AbnormalMode`	Abnormal mode status appears if the current operating mode is not normal mode.	*Bit1*
`AbnormalInterlockBy-pass`	Abnormal interlock bypass condition appears if the Interlock bypass is enabled.	*Bit2*
`AbnormalOperatorTab`	Abnormal status of operator tab. This condition appears if the Interlock bypass is enabled.	*Bit8*
`AbnormalFailTab`	Abnormal status of **Failures** tab.	*Bit10*
`AbnormalEngineeringTab`	Abnormal status of engineering tab.	*Bit12*
`AbnormalDirection1In-terlockTab`	Abnormal status of direction 1 interlock tab.	*Bit13*
`AbnormalDirection2In-terlockTab`	Abnormal status of direction 2 interlock tab.	*Bit14*

Action Required Conditions

The table describes the action required conditions of variable speed motor:

Item name	Description	Address
Bit items derived from ActionRequired variable word.		
`ActionOperatorTab`	Action required status of operator tab.	*Bit8*
`ActionFailTab`	Action required status of **Failures** tab.	*Bit10*
`ActionDirection1In-terlockTab`	Action required status of direction 1 interlock tab.	*Bit13*
`ActionDirection2In-terlockTab`	Action required status of direction 2 interlock tab.	*Bit14*

Alarms

The table describes the advanced alarm items that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
`MotMNTOpera-tionHoursAlm`	Running hours counter overflow	`GPL_BitValueCheck (StatusWord, 0)`
`MotMNTManeuver-sNumbersAlm`	Operations counter overflow	`GPL_BitValueCheck (StatusWord, 1)`
`Alarm`	Detection of unsuccessful operation	`GPL_BitValueCheck (StatusWord, 4)`
`ILCKTrip`	Interlock trip	`GPL_BitValueCheck (StatusWord1, 3,4,0)`
`Fail`	Inoperable variable speed motor	`GPL_BitValueCheck (StatusWord1, 11)`

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
StateAlarm	All changes in the variable speed motor status and output.
ModeAlarm	All changes in the mode of operation of the equipment.
ActionAlarm	All the operator actions in the supervision.

Trends

The table describes the trend items that are used by Supervision components for variable speed motor:

Item name	Expression	Description
PV	MOTORVSGP_ST.PV	Present value
SP	MOTORVSGP_ST.SP	Setpoint
OP	MOTORVSGP_ST.OP	Output value

Valves

What's in This Part

Valve 229
Motorized Control Valve 254
Hand Valve 283
Dual Output Valve 294
Motorized Valve 324
Control Valve 352

Overview

This part provides the functionality of the templates, a detailed description of the control (functions, pins, pin layout, and variables of the function blocks), and the supervision functions of valves.

These components do not reflect any specific installation.

⚠ WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate them.
- Review the implications of communication link interruptions and take actions to mitigate them.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, and applicable codes and regulations.
- Apply local accident prevention and safety regulations and guidelines.[1]
- Test each implementation of a system for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

Valve

What's in This Chapter

Template 229
Control 233
Optional Service - Local Panel (VALVELPGP) 240
Supervision 244

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the valve.

Template

Overview

The `$ValveGP` process control module template is used to manage single actuator devices, such as single actuator valves.

Description

General Description

The `$ValveGP` process control module template is used to control a digital signal usually associated with a physical output.

The process template allows managing the associated devices from the sequential control, the continuous/trigger based control, and/or the supervision system. It depends on their configuration and the system needs.

Function Description

The main functions of the valve template are described in the following table:

Function	Description
Owner selection	The object manages the control system level, which is the owner (operator or program). As a result, it is responsible for setting the set-point for the desired position.
Local and remote set-point	Enables to control the template with a set-point that is determined through a set-point selector (local or remote) in program owner. The local set-point is assigned from the sequential control while the remote set-point is assigned to the template from the logic implemented in the continuous/trigger based control.
Running status	Allows to determine the running status of the element to be controlled and monitored with the help of open and close feedback.
Interlocking	The template gives a command to the device to move to the defined position in case of an active interlock. An interlock bypass and Setpoint is reset in case of interlock function is available.
Mode selection	The object can be switched to different modes of operation like Simulation, Maintenance, Out Of Service and Override.
Inoperable device	Allows to monitor inoperable device condition (optional). Template can be configured to withdraw output based on this condition and setpoint can be withdrawn in case of detected error.

Function	Description
Detection of unsuccessful operation	If position is unsuccessful within a configured time after the output command is activated, template detects unsuccessful operation.
Manual resetting	Allows you to reset manually after the detection of an inoperable device condition or unsuccessful operation. After reset, device output will follow set point. It also allows to reset all the rearm conditions of optional services as well on specific configuration.

Optional Services

Available optional services are:

- **Interlocks**
- **Local Panel**
- **Maintenance**

Parameters

Parameters

The `$ValveGP` templates provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default value	Additional remarks
Setpoint	**Reset on interlock**	`Boolean`	*FALSE*	Resets the interlock when the motor is interlocked.
	Track setpoint on interlock	`Boolean`	*TRUE*	The interlock setpoint is tracked by the equipment setpoint when the interlock condition is active.
Failure	**Confirmation timeout**	`Duration`	*00:00:-03*	Maximum time for feedback confirmation.
	Minimum alarm duration		*00:00:-05*	Minimum time to maintain the confirmation alarm or detected error alarm.
Rearm	**Required on failure**	`Boolean`	*TRUE*	Device requires a reset after device error or interlock trip alarm.
	Required on confirmation failure			Device requires a reset after confirmation error.
	Rearm resets conditions		*FALSE*	Rearms the valve, interlock and detected error conditions.

Inputs (Digital Input)

The table describes the digital input:

Name	Description
`OpenPosition`	It allows you to connect the open feedback of the valve.
`ClosePosition`	It allows you to connect the closed feedback of the valve.
`ExternalControl`	It allows you to control the device externally.

For more details, refer to the topic describing the parameters of digital inputs, page 44.

Outputs (Digital Output)

The table describes the digital output:

Name	Description
OP	It allows you to connect the digital output signal of the valve to external device.

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Local Panel (Logic)

For more details, refer to the topic describing the parameters of logic, page 45.

Maintenance

For more details, refer to the topic describing the parameters of maintenance, page 89.

Local Panel (Digital Input)

The table describes the digital input:

Name	Description
LPMode	Local panel selection signal from mode selector switch.
ZEROMode	Zero/Off selection signal from mode selector switch.
CSMode	Control selection signal from selector switch.
OffPB	Motor start signal from local panel.
OnPB	Motor stop signal from local panel.

For more details, refer to the topic describing the parameters of digital inputs, page 44.

Local Panel (Digital output)

The table describes the digital output:

Name	Description
PBLight	Output signal to indicate that push buttons are operational.

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default value	Additional remarks
Configu-ration	**Enable Separate Alarms for Open and Close Confirmation Failure**	Boolean	*FALSE*	It enables separate alarms for open and close detected confirmation error.
Operation	**Normal Owner**	Enum	*Program and Cascade (5)*	It allows to define the normal owner groups and any other

Section	Parameter description	Type	Default value	Additional remarks
				owner who has logged in indicates an abnormality in graphics.
Security	**Set Owner**	Enum	*Operator (1)*	Privilege to change owner. This parameter is used in Operator Tab, page 247.
	Set Setpoint		*Operator (1)*	Privilege to change setpoint. This parameter is used in Operator Tab, page 247.
	Set Mode - Override		*Supervisor (2)*	Privilege to change override mode. This parameter is used in Engineering Tab, page 248.
	Set Mode - Simulation		*Engineer (5)*	Privilege to change simulation mode. This parameter is used in Engineering Tab, page 248.
	Set Mode - Maintenance		*Maintenance (3)*	Privilege to change maintenance mode. This parameter is used in Engineering Tab, page 248.
	Set Mode - Out of Service		*Supervisor (2)*	Privilege to change out of service mode. This parameter is used in Engineering Tab, page 248.
	Set Reset		*Operator (Confirmed) (10)*	Privilege to reset. This parameter is used in Operator Tab, page 247.
	Set Alarm Acknowledge		*Operator (1)*	Privilege to acknowledge alarm. This parameter is used in Alarm Tab, page 67.
	Set Bypass Conditions		*Supervisor (2)*	Privilege to set bypass conditions mode. This parameter is used in Engineering Tab, page 248.
	Set Bypass All Interlocks		*Supervisor (2)*	Privilege to bypass all interlocks. This parameter is used in Engineering Tab, page 248.
	Set Asset Maintenance		*Operator (1)*	Privilege to set maintenance mode. This parameter is used in Engineering Tab, page 248.
Events	**Enable log**	Boolean	*FALSE*	Enable event logging for asset status. Example: Opening
Historize	**Owner/Mode**	Boolean	*FALSE*	It allows to historize the change in mode and owner.

Asset and Display Parameters

For more details, refer to the topic describing the parameters of asset and display, page 43.

Alarm Parameters

For more details, refer to the topic describing the parameters of alarm, page 45.

Local Panel

For more details, refer to the topic describing the parameters of local panel, page 44.

Maintenance

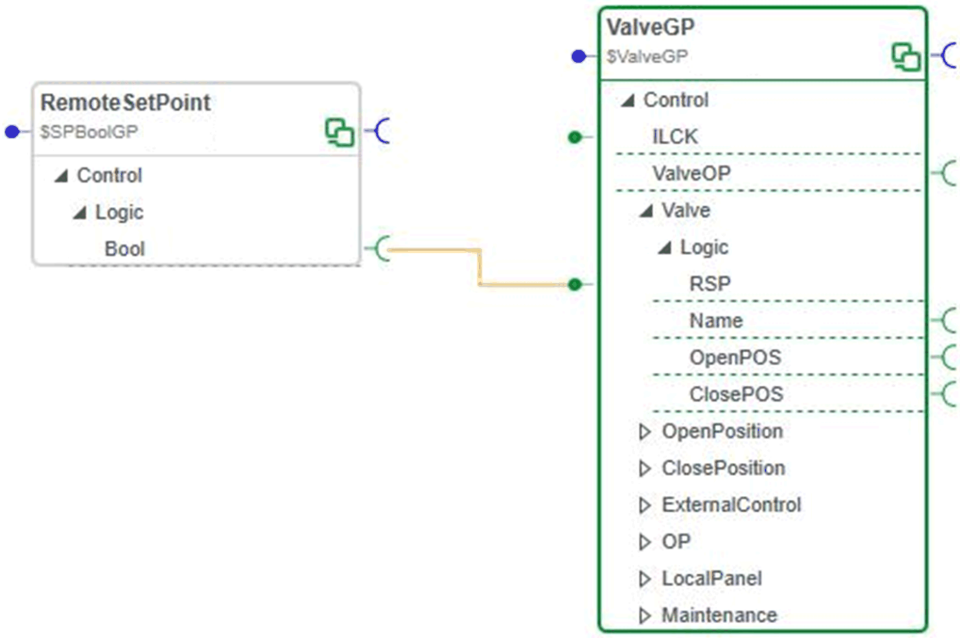
For more details, refer to the topic describing the parameters of maintenance, page 89.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

The following figure depicts the `$ValveGP` (also applicable for `$ValveCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



For interface identifiers, refer to the control module template interfaces, page 49.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

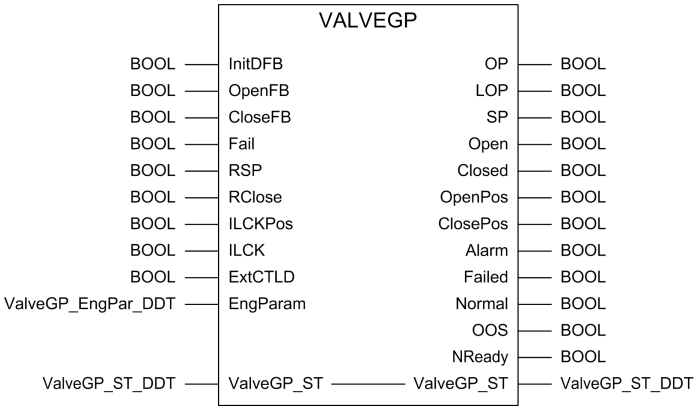
Overview

This section describes the `VALVEGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Name	Type	Description
`InitDFB`	`BOOL`	If a rising edge is detected on this input, the internal timers of the block are reset and detected alarms are set to 0.
`OpenFB`	`BOOL`	1 = Indicates that the open valve position is reached. Used to connect to the digital input of the open-valve limit switch. Disabled internally by the DFB if no variable or value is connected.
`CloseFB`	`BOOL`	1 = indicates that the closed valve position is reached. Used to connect to the digital input of the closed-valve limit switch. Disabled internally by the DFB if no variable or value is connected.
`Fail`	`BOOL`	1 = Indicates to the DFB a detected error in the valve.
`RSP`	`BOOL`	Remote setpoint by continuous control (0 = Close, 1 = Open). If `RClose` is connected, then the rising edge (Pulse based control) sends the command to open the valve only when program/remote cascade mode is configured.
`RClose`	`BOOL`	Remote close setpoint (Rising edge to close valve).
`ILCKPos`	`BOOL`	1 = Position needs to be used when the DFB is interlocked. Refer to the `ILCK` input.
`ILCK`	`BOOL`	1 = Interlocks the valve at the defined position. **NOTE:** If the valve requires resetting (`STW.Reamr` = 1), the `OP` takes the value = 0 regardless of the `ILCKPos` value waiting for the operator to carry out the reset.
`ExtCTLD`	`BOOL`	1 = Indicates to the DFB whether the device is being controlled from outside of the system. Monitoring of confirmation alarms is stopped during external control. The error input continues to be evaluated.
`EngParam`	`ValveGP_EngPar_DDT`	Engineering parameters

ValveGP_EngPar_DDT Type

Name	Type	Description
TimeOut	TIME	Enables to configure the maximum time that the devicetakes to reached defined position. When the output is triggered, if the limit switch is not detected even after the TimeOut time is elapsed. The device is switched to detected alarm state if the limit switch is not detected. When it has a value of 0, monitoring is deactivated.
ScanTime	TIME	The minimum time for which the detected alarm signals are kept active. ScanTime confirms that the supervision system acquires the data for detected alarms that are automatically reset.
ConfRearmEn	BOOL	• 1 = Withdraws the output *OP* when a non-confirmed operation is detected (*Alarm* = 1) and requires a mandatory manual resetting once the *Alarm* condition disappears. • 0 = The output *OP* remains in its previous state when a non-confirmed operation is detected (*Alarm* = 1) and does not need manual resetting once the *Alarm* condition disappears. Refer the TimeOut input and the ValveGP_ST.STW.Alarm input/output.
FailRearmEn	BOOL	• 1 = A manual resetting is required when the detected error/interlock trip alarm condition disappears. • 0 = No manual resetting is required when the detected error/interlock trip alarm condition disappears.
ResetSPFail	BOOL	Reset setpoint in case of detected error condition (1 = Re-initiation of command is required if valve closes due to detected error or alarm).
TrackSPEn	BOOL	Track set-point in case of that the interlock condition is active. **NOTE:** The *TrackSPEn* parameter enables the interlock trip alarm when the device stopped due to interlock conditions.

Outputs

Output Parameter Description

Name	Type	Description
OP	BOOL	1 = Field Command signal active. Allows the control signal of the device to be connected. It always follows the LOP command. **NOTE:** During simulation the signal will be deactivated.
LOP	BOOL	1 = Logical command signal active. Refer to the LOP Calculation, page 236.
SP	BOOL	1 = Current device set-point active. Refer to the SP Calculation, page 236.
Open	BOOL	1 = Indicates the maximum opening position is reached. It is calculated based on the states obtained from the analog input and/or the digital input belonging to the **open-valve limit** switch. It is evaluated by considering other detectors. Refer to the output Closed and input/output ValveGP_ST.Sim, page 237. Refer to the Open and Closed feedback calculation, page 237.
Closed	BOOL	1 = Indicates the closed-valve position is reached. It is calculated based on the states obtained from the analog input and/or digital input belonging to the **closed-valve limit** switch.

Name	Type	Description
		It is evaluated by considering other detectors. Refer to the output Open and input/output ValveGP_ST.Sim, page 237. Refer to the Open and Closed feedback calculation, page 237.
OpenPos	BOOL	1 = Indicates the maximum opening position is reached. It is calculated based on the states obtained from the analog input and/or the digital input belonging to the open valve limit switch. It is independent from the other detector if the CloseFB detector is available. Refer to the OpenPos and ClosePos feedback calculation, page 237.
Close-Pos	BOOL	1 = Indicates the closed valve position is reached. It is calculated based on the states obtained from the analog input and/or the digital input belonging to the close valve limit switch. It is independent from the other detector if the OpenFB detector is available. Refer to the OpenPos and ClosePos feedback calculation, page 237.
Alarm	BOOL	1 = The detection of unsuccessful operation has occurred.
Failed	BOOL	1 = Indicates an inoperable device. It is evaluated according to the value of the Fail and if FailRearmEn engineering parameter configured then Failed will be high until Rearm is provided. When it is activated, it is timed so that the signal is maintained for a minimum of ScanTime. 0 = After the minimum activation time has passed, it is only deactivated if the Fail input signal is deactivated.
Normal	BOOL	Indicates the valve is operating in normal mode.
OOS	BOOL	1 = Indicates the device is out of service.
NReady	BOOL	Indicates valve not ready for operation.

LOP Calculation

The evaluation of the device control depending on the interlock status, the current set-point, and if the DFB requires resetting is shown in the following table:

STW.RearmR	SC.ILCKD	ILCKPos	LOP is calculated as:
ON	-	-	OFF
OFF	ON	ON	ON
OFF	ON	OFF	OFF
OFF	OFF	-	ValveGP_ST.STW.SP

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

The *ConfRearmEn* engineering parameter has to be set to 1 for the *OP* pin of the function block output to become 0, under non-confirmed operations.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: Non-confirmed operations are those for which commands have been given and to which there is no confirming reply from the equipment.

SP Calculation

The signal evaluation depending on the SC.Rem and RSP inputs and the ValveGP_ST.CFGW.Owner input/output is shown in the following table:

Owner	Rem	SP is calculated as
OFF	OFF	Sequential control (SC.LSP) or Pulse base control (SC.LStart, SC.LStop)
OFF	ON	Continous control (RSP) or Pulse base control if RClose pin is connected (RSP, RClose)
ON	-	ValveGPST.CFGW.LSP

Feedback Calculation for Open and Closed

OpenFB connected	CloseFB connected	Simulation	Override	Open is calculated as:	Closed is calculated as:
-	-	-	ON	LOP	Not LOP
-	-	ON	OFF	LOP and (TimeOut/2)	Not LOP and (TimeOut/2)
OFF	OFF	OFF	OFF	LOP	Not LOP
OFF	ON	OFF	OFF	Not CloseFB	CloseFB
ON	OFF	OFF	OFF	OpenFB	Not OpenFB
ON	ON	OFF	OFF	OpenFB and (not CloseFB)	CloseFB and (not OpenFB)

Feedback Calculation for OpenPos and ClosePos

OpenFB connected	CloseFB connected	Simulation	Override	OpenPos is calculated as:	ClosePos is calculated as:
-	-	-	ON	LOP	Not LOP
-	-	ON	OFF	LOP and (TimeOut/2)	Not LOP and (TimeOut/2)
OFF	OFF	OFF	OFF	LOP	Not LOP
OFF	ON	OFF	OFF	Not CloseFB	CloseFB
ON	OFF	OFF	OFF	OpenFB	Not OpenFB
ON	ON	OFF	OFF	OpenFB	CloseFB

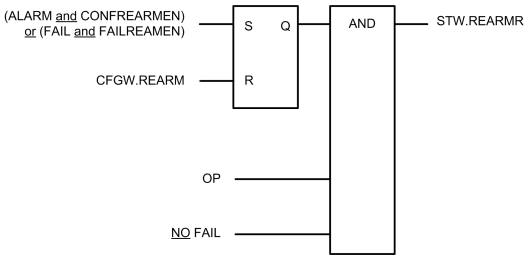
Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
ValveGP_ST	ValveGP_ST_DDT	Data structure of the DFB status.

ValveGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	OpenPos	*Bit0*	BOOL	Refer to OpenPos output pin, page 235.
	ClosedPos	*Bit1*	BOOL	Refer to ClosedPos output pin, page 235.

Name	Bit Name	Bit	Type	Description
	SP	*Bit2*	BOOL	Refer to SP output pin, page 235.
	LOP	*Bit3*	BOOL	Refer to LOP output pin, page 235.
	Alarm	*Bit4*	BOOL	Refer to Alarm output pin, page 235.
	Failed	*Bit5*	BOOL	Refer to Failed output pin, page 235.
	Rem	*Bit6*	BOOL	• 1 = Enables the DFB to be configured to remote set-point Rem. • 0 = Enables the DFB to be configured to local set-point LSP.
	ILCK	*Bit7*	BOOL	Refer to ILCK input pin, page 234
	RearmR	*Bit8*	BOOL	Indicates if the device requires resetting (1) after the detection of an inoperable device condition if it has been configured through the ConfRearmEn and/or FailRearmEn inputs. The signal activation logic included in the DFB is as follows. 
STW	ExtCTLD	*Bit9*	BOOL	Read-only access Refer to ExtCTLD input pin, page 234
	NReady	*Bit10*	BOOL	Refer to NReady output pin, page 235.
	OpenAlarm	*Bit11*	BOOL	1 = Valve open position confirmation alarm.
	CloseAlarm	*Bit12*	BOOL	1 = Valve close position confirmation alarm.
	SPLocked	*Bit13*	BOOL	Indicates whether the set-point change buttons in the monitoring (HMI) system should be locked (1) or not (0). Is activated when: • The Local Panel is set to zero mode. • The local panel is set to local panel mode and the LockSetpointEn input signal has been configured as active (1).
	ILCKTrip	*Bit15*	BOOL	1 = Interlock trip active.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read-only access to the data contained in this word.
	Owner	*Bit0*	BOOL	Read/write access Refer to Owner, page 239.
	ILCKBP	*Bit1*	BOOL	Read/write access 1 = Enables the interlock to be bypassed.
	Override	*Bit2*	BOOL	Refer to Override, page 239.
	Rearm	*Bit3*	BOOL	Write access Refer to Rearm, page 239.
	LSP	*Bit4*	BOOL	Read/write access Refer to LSP, page 239.
	SimFBFail	*Bit12*	BOOL	1 = Feedback will not simulate in simulation mode.
	Sim	*Bit13*	BOOL	Read/write access Refer to Sim, page 239.
	Maint	*Bit14*	BOOL	Refer to Maint, page 239.
	OOS	*Bit15*	BOOL	Read/write access Refer to OOS output pin, page 235.

Public Variables

Public Variable Description

Variable	Type	Description
SC	ValveGP_SC_DDT	Provides the frequently needed data to monitor the device status and to control it from the sequential control.

ValveGP_SC_DDT Type

Name	Type	Description
Rem	BOOL	Read/write access 1 = Enables the DFB to be configured to remote set-point RSP 0 = Enables the DFB to be configured to local set-point LSP.
LSP	BOOL	Read/write access 1 = Enables the assignment of target local set-point usually for the sequential control if the owner is the Program (Owner input/output is OFF) and the selected set-point is Local (SC.Rem public variable is OFF). 0 = The block automatically assigns the current value of the resulting set-point. It is described as follows:

Owner	Rem	SC.LSP
OFF (Program)	OFF	SC.LSP (last value or new value assigned from the sequential control)
OFF	ON	SP (calculated from the RSP)
ON (Operator)	ON	SP (last value or new value assigned by the Operator)

Name	Type	Description
LOpen	BOOL	Read/write access 1 = Rising edge to enable Open command while the owner is in Program and Local.
LClose	BOOL	Read/write access 1 = Rising edge to enable Close command while the owner is in Program and Local.
OP	BOOL	Read-only access Refer to OP output pin, page 235.
LOP	BOOL	Read-only access Refer to LOP output pin, page 235.
SP	BOOL	Read-only access Refer to SP output pin, page 235.
Open	BOOL	Read-only access Refer to Open oitput pin, page 235.
Closed	BOOL	Read-only access Refer to Closed pin, page 235.
OpenPos	BOOL	Read-only access Refer to OpenPos output pin, page 235.
ClosePos	BOOL	Read-only access Refer to ClosePos output pin, page 235.

Name	Type	Description
Alarm	BOOL	Read-only access Refer to Alarm output pin, page 235.
Failed	BOOL	Read-only access Refer to Failed output pin, page 235.
ILCKD	BOOL	Read-only access The signal evaluation depending on the ILCK input and the ValveGP_ST.CFGW.ILCKBP input/output is shown as follows: **ILCKD** / **ILCKBP** / **SC.ILCKD** OFF / OFF / OFF OFF / ON / OFF ON / OFF / ON ON / ON / OFF
Rearm	BOOL	Read/write access Allows the device to be reset (1). The block sets the signal to 0 after each execution and sets the signal indicating that the resetting is required (CFGW.Rearm) to 0.
RearmR	BOOL	Read/write access Refer to RearmR input/output pin, page 237.
Owner	REAL	Read/write access Enables to configure whether the set-point is set by the Program (0) or by the Operator (1).
Normal	BOOL	Read-only access Refer to Normal output pin, page 235.
Override	BOOL	Read/write access Enables to determine if the DFB is operating with external output (1 = Override) or not (0 = Normal).
Sim	BOOL	Read-only access Enables the simulation mode.
Maint	BOOL	Read-only access Maintenance mode is active).
OOS	BOOL	Read/write access Refer to OOS output pin, page 235.
NReady	BOOL	Read-only access Refer to NReady output pin, page 235.
ExtCTLD	BOOL	Read-only access Refer to the ExtCTLD input pin, page 234.

Optional Service - Local Panel (VALVELPGP)

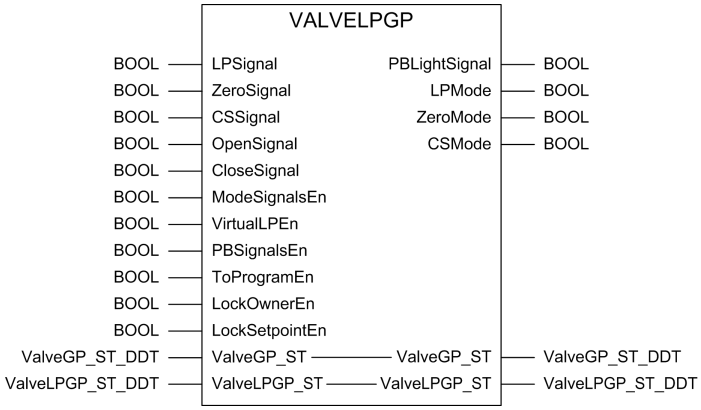
Overview

This chapter describes the optional services - local panel (VALVELPGP) DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Name	Type	Description
`LPSignal`	`BOOL`	1 = The local panel mode switch is in the Local Panel position. Refer to the table included with the description of the `LPMode` output pin, page 242.
`ZeroSignal`	`BOOL`	1 = The local panel mode switch is in the zero position.This signal is optional even if a mode switch is available on the local panel. When this signal is a logical high, user is not allowed to operate the device from the faceplate and also from the DFB. Hence, all the functions in the faceplate will be disabled. Refer to the table included with the description of the `LPMode` output pin, page 242.
`CSSignal`	`BOOL`	1 = The local panel mode switch is in the Control System position. Refer to the table included with the description of the `LPMode` output pin, page 242.
`OpenSignal`	`BOOL`	1 = Indicates to the DFB that `Open` push button on the local panel is pressed.
`CloseSignal`	`BOOL`	1 = Indicates to the DFB that `Close` push button on the local panel is pressed.
`ModeSignalsEn`	`BOOL`	1 = Enables the use of the Local Panel/Zero/Control System (or Local Panel - Control System) mode switch on the local panel. Usually configured in development phase based on the characteristics of the local panel being used. If the mode switch is enabled, the `LPSignal`, `ZeroSignal`, and `CSSignal` inputs are considered to determine the operating modes of the local panel. Refer to the table included with the description of the `LPMode` output pin, page 242. **NOTE:** *ModeSignalsEn* input pin has higher priority than the *VirtualLPEn* input pin signal.
`VirtualLPEn`	`BOOL`	1 = This signal enables the operator to select the Local/Control System mode of operation from the HMI, so that the push button signals are enabled for operation. **NOTE:** When the owner is program, the Local/Control System mode selection drop-down list will be visible but disabled for operation however, when the owner is operator the Local Panel drop-down list will be accessible for operation. Refer to the table included with the description of the `LPMode` output pin, page 242.
`PBSignalsEn`	`BOOL`	1 = Enables to accept the OFF and ON push button signals from the local panel. This signal is not applicable if the mode switch is enabled (refer to the `ModeSignalsEn` input). It means that push button signals are considered when in Local Panel mode. When the mode switch is disabled (`ModeSignalsEn` = 0), the `PBSignalsEn` signal enables/disables the push button signals from the control system itself based on the relevant process conditions. The `SC.DisableLP` public variable is also considered for determining whether the push buttons are enabled or not when *LPMode* pin signal is high, as shown in the following table.

Vir-tualLPEn	ModeSignalsEn	PBSignalsEn	SC. DisableLP	Push buttons enabled?
OFF	OFF	OFF	-	NO
OFF	OFF	-	ON	NO

Name	Type	Description
		OFF | OFF | ON | OFF | YES OFF | ON | - | - | YES ON | OFF | OFF | - | YES Therefore, this signal can be used to enable/disable the push buttons on a simple local panel that does not feature a mode switch (Local Panel/Zero/Control System) based on relevant process conditions.
ToProgramEn	BOOL	1 = Change to program mode when control is switched to control system mode and VirtualLPEn = 0.
LockOwnerEn	BOOL	1 = Disables the access of changing the owner from HMI while the mode switch is in local panel. The following table shows how the on-off device owner is evaluated based on *ToProgramEn*, *LockOwnerEn* input signals and the local panel operating mode: **Local Panel Mode** | **ToProgramEn** | **LockOwnerEn** | **Owner** Zero mode | - | - | Operator Switch to Local Panel | - | - | Operator Local Panel | - | ON | Operator Switch to Control System | ON | - | Program In remaining cases, owner of the on/off device ceases to be defined from the ValveLPGP DFB and it enables to change it from monitoring (HMI) system with the command word in ValveLPGP_ST.CFGW input/output of the corresponding ValveLPGP DFB. In any case, consider that, the device owner that had been set before switching to the local panel mode is not memorized.
LockSetpointEn	BOOL	1 = Disables the access to set point of on/off devices in the monitoring (HMI) system when it is in local panel mode only, and the owner is operator. Therefore, user can activate this signal to lock setpoint operation from the monitoring (HMI) system while on/off device is controlled from the local panel.

Outputs

Output Parameter Description

Name	Type	Description
PBLightSignal	BOOL	1 = The push buttons on the local panel are fully operational, that is, the local panel is in Local Panel mode (either because the mode is selected with the corresponding switch or because the push buttons are enabled in the event that there is no selector switch available or the *VirtualLPEn* is high with the local panel mode enabled).
LPMode	BOOL	1 = The local panel mode is in Local Panel. The following table shows how the local panel operating mode is determined based on the ModeSignalsEn, VirtualLPEn, PBSignalsEn, LPSignal, CSSignal and ZeroSignal, input signals: **ModeSignalsEn** | **VirtualLPEn** | **PBSignalsEn** | **LPSignal** | **CSSignal** | **ZeroSignal** | **OUTPUT** 0 | 0 | 0 | 0 | 0 | 0 | CS mode 1 | 0 | 0 | 0 | 0 | 0 | Zero mode 1 | 0 | 0 | 1 | 0 | 0 | LP mode and PBLightSignal 1 | 0 | 0 | 0 | 1 | 0 | CS mode 0 | 1 | 1 | NA | NA | NA | CS mode 0 | 1 | 0 | NA | NA | NA | CS mode 0 | 0 | 1 | 0 | 0 | 0 | LP mode and PBLightSignal 1 | 1 | 0 | 1 | 0 | 0 | LP mode and PBLightSignal

Name	Type	Description						
		1	1	0	0	0	0	Zero mode
		1	1	0	0	0	1	Zero mode
		1	1	0	0	1	0	CS mode
		1	0	1	0	1	0	CS mode
		1	1	1	0	1	0	CS mode
		1	0	0	0	0	1	Zero mode
ZeroMode	BOOL	1 = The local panel mode is in Zero. User is not allowed to operate the device from the faceplate and also from DFB. Hence, all the functions in the faceplate will be disabled. Refer to the table included with the description of the LPMode output.						
CSMode	BOOL	1 = The local panel mode is in Control System. Refer to the table included with the description of the LPMode output.						

Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
ValveGP_ST	ValveGP_ST_DDT, page 237	Data structure corresponding to the on-off device is to be controlled from the local panel (for detailed information regarding this structure, refer to the ValveGP DFB, page 234. The structure provides the information needed for the operation of the ValveLPGP DFB and enables this DFB to control the owner and set-point for the on-off device.
ValveLPG-P_ST	ValveLPGP_ST_DDT	Data structure is used as an interface with the monitoring (HMI) system.

ValveLPGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	LPMode	*Bit0*	BOOL	Local panel in Local Panel mode. Refer to the LPMode output pin, page 242.
	ZeroMode	*Bit1*	BOOL	Local panel in zero mode. Refer to the ZeroMode output pin, page 242.
	CSMode	*Bit2*	BOOL	Local panel in Control System mode. Refer to the CSMode output pin, page 242.
	OwnerLocked	*Bit3*	BOOL	Indicates whether the owner change buttons in monitoring (HMI) system needs to be locked (1) or not (0). Is activated when: • The Local Panel is set to zero mode. • The local panel is set to local panel mode and the *LockOwnerEn* input signal has been configured as active (1).
	SPLocked	*Bit4*	BOOL	Indicates whether the set-point change buttons in the monitoring (HMI) system should be locked (1) or not (0). Is activated when: • The Local Panel is set to zero mode. • The local panel is set to local panel mode and the *LockSetpointEn* input signal has been configured as active (1).
	ModeSignalsEn	*Bit5*	BOOL	Indicates the state of the ModeSignalsEn input signal, that is, whether there is a mode switch (1) or not (0).
	PBEnabled	*Bit6*	BOOL	Indicates whether the push buttons of the local panel are enabled (1) or not (0). The signal is activated when at least one of the ModeSignalsEn and PBSignalsEn signal inputs is active.

Name	Bit Name	Bit	Type	Description
	VirtualLPEn	*Bit7*	BOOL	Enables the drop-down list for the Local Panel/Control System mode selection.
	LPSignal	*Bit8*	BOOL	Indicates the state of the LPSignal input signal.
	ZeroSignal	*Bit9*	BOOL	Indicates the state of the ZeroSignal input signal.
STW	CSSignal	*Bit10*	BOOL	Indicates the state of the CSSignal input signal.
	CloseSignal	*Bit11*	BOOL	Indicates the state of the CloseSignal input signal.
	OpenSignal	*Bit12*	BOOL	Indicates the state of the OpenSignal input signal.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read-only access to the data contained in this word.
	LPEnabled	*Bit0*	BOOL	Read/write access 1 = Local Panel enable command from supervision.

Public Variables

Public Variable Description

Variable	Type	Description
SC	ValveLPGP_SC_DDT	Provides the frequently needed data for monitoring and controlling the DFB.

ValveLPGP_SC_DDT Type

Name	Type	Description
LPMode	BOOL	Read-only access Refer to the LPMode output pin, page 242.
ZeroMode	BOOL	Read-only access Refer to the ZeroMode output pin, page 242.
CSMode	BOOL	Read-only access Refer to the CSMode output pin, page 242.
DisableLP	BOOL	Read/write access 1 = Disables the push buttons on the local panel. Only applicable when a local panel without mode switch configuration is used (ModeSignalsEn = 0).

Supervision

Overview

This section describes the Supervision functions and runtime services that are available for the valve.

Genies

Genie Properties

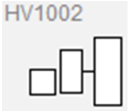
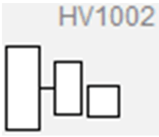
Refer to Genie Properties, page 62

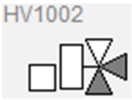
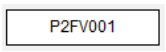
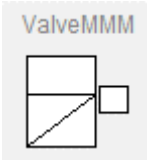
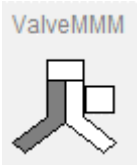
Genie Icons

Refer to Genie Icons, page 63

Representation

The table describes the genies of the `gpl_valve` library:

Graphic symbol	Genie name	Description
	`Butterfly_Vert_Left`	Vertical butterfly with control on the left.
	`Butterfly_Vert_Right`	Vertical butterfly with control on the right.
	`2way_Vert`	Vertical two-way valve.
	`2way_Horiz`	Horizontal two-way valve.
	`3way_Horiz_Down_Left`	Horizontal three-way valve (down to left way when closed, down to right way when open).
	`3way_Horiz_Down_Right`	Horizontal three-way valve (down to right way when closed, down to left way when open).
	`3way_Horiz_Left_Down`	Horizontal three-way valve (left to down way when closed, left to right way when open).
	`3way_Horiz_Left_Right`	Horizontal three-way valve (left to right way when closed, left to down way when open).

Graphic symbol	Genie name	Description
	`3way_Horiz_Right_Down`	Horizontal three-way valve (right to down way when closed, right to left way when open).
	`3way_Horiz_Right_Left`	Horizontal three-way valve (right to left way when closed, right to down way when open).
	`3way_Vert_Down_Right`	Vertical three-way valve (down to right way when closed, down to up way when open).
	`3way_Vert_Down_Up`	Vertical three-way valve (down to up way when closed, down to right way when open).
	`3way_Vert_Right_Down`	Vertical three-way valve (right to down way when closed, right to up way when open).
	`3way_Vert_Right_Up`	Vertical three-way valve (right to up way when closed, right to down way when open).
	`3way_Vert_Up_Down`	Vertical three-way valve (up to down way when closed, up to right way when open).
	`3way_Vert_Up_Right`	Vertical three-way valve (up to right way when closed, up to down way when open).
	`valveinstanceid_m`	Valve instance.
	`damper_m`	Valve - damper.
	`diverter_m`	Valve - diverter.

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

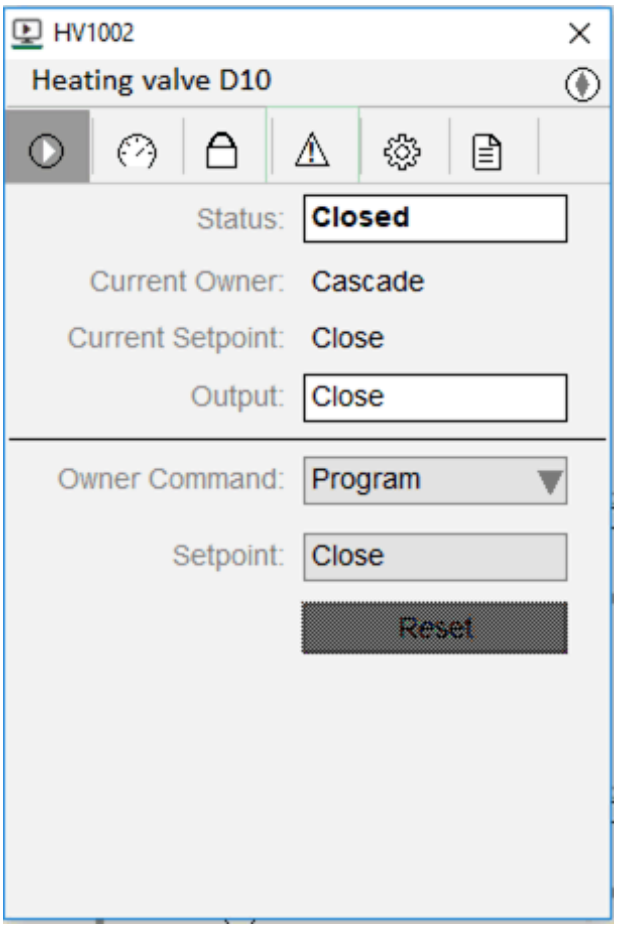
Available Tabs

During operation, clicking a valve genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
  - Event, page 69
- Tabs for optional functions, which appear only if configured:
  - Interlock, page 87
  - Measures, page 70

Operator Tab

The figure shows an operator tab:

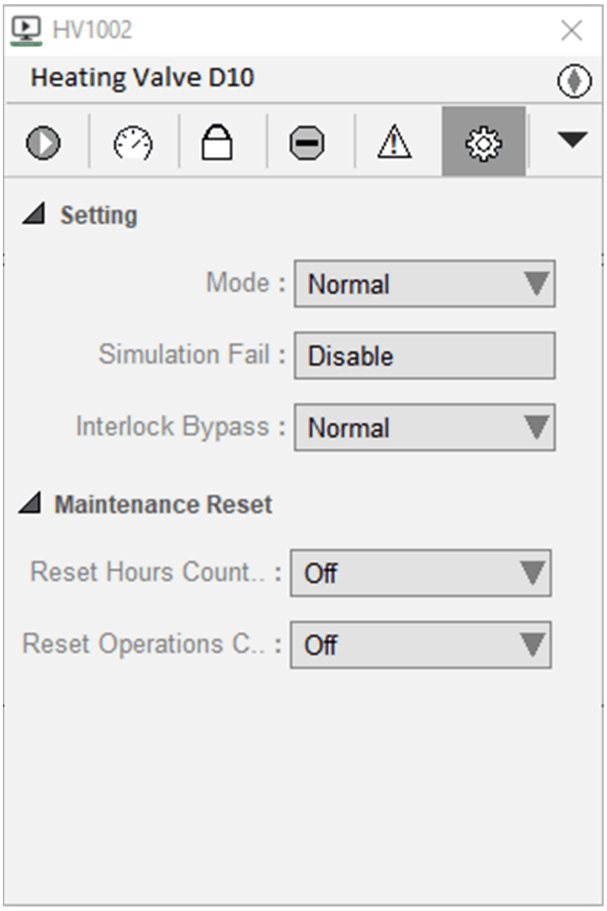


The table describes the control function of an operator tab:

Label	Item name	Description	Security parameter, page 141
Status	EquipStatus	Equipment status of valve, refer to EquipStatus, page 159.	—
Current Owner	Owner	Current owner of valve, refer to Owner, page 58.	
Current Setpoint	Setpoint	Current setpoint of valve.	
Output	EquipState	Output status of valve, refer to EquipState, page 159.	
OwnerCom-mand	OwnerCommand	Owner command of valve (Program, Operator or Local Panel). **NOTE:** Local panel owner is enabled when the virtual panel is selected.	Set owner
Setpoint	CommandSet-point	Allows you to change the setpoint, if owner = operator.	Set setpoint
–	Rearm	Allows you to reset manually after the detection of an inoperable device condition or unsuccessful operation. **NOTE:** When the control module is reset, the current setpoint shown in operator tab is effective. Also, when you click **Reset**, a dialog box opens to confirm.	Set rearm

Engineering Tab

The figure shows an engineering tab:



The table describes the control function of engineering tab:

Label	Item name	Description	Security parameter, page 231
Setting			
Mode	Mode	Allows you to view and change the mode. For more details, refer to operating mode, page 59.	Set mode - override, set mode - simulation, set mode - maintenance, and Set mode - out of service **NOTE:** Each configurable mode has the different security parameters
Simulation Fail	SimulationFailActive	Enable or disable feedback simulation. **NOTE:** The Simulation Fail will only be available when the parameter `SimulationFail` is created in the citect parameters. To create, go to **Setup** > **Parameters**. Enter the `GPL.Mode` in **Section Name**, `SimulationFail` in **Name** and set the **Value** = 1.	Set mode - simulation
Interlock Bypass	InterlockBypass	Interlock bypass status Allows you to bypass interlock.	Set bypass conditions, set bypass all interlocks
Maintenance Reset			
Reset Hours Count	VlvMntResetHour-sCounter	Reset the hours counter.	Set reset
Reset Operations Count	VlvMntResetManeuver-sCounter	Reset the operations counter.	Set reset

Items

Overview

This section describes the variables, equipstatus and equipstate variable, abnormal and action required conditions, alarms, events, and trends items of `valve`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`FeedbackOpen`	Open feedback	Open/Not Open	*Bit0*
`FeedbackClosed`	Closed feedback	Closed/Not Closed	*Bit1*
`Setpoint`	Selected setpoint	On/Off	*Bit2*
`Output`	Output status		*Bit3*
`AlarmActive`	Feedback confirmation detected error	Active/Inactive	*Bit4*
`FailActive`	Valve inoperable		*Bit5*
`ProgramSetpoint-Mode`	Setpoint source for program owner	Remote/Sequence	*Bit6*
`InterlockActive`	Interlock status	Interlock/Healthy	*Bit7*
`RequireRearm`	Reset required	Required/Not Required	*Bit8*

Item name	Description	Enumeration	Address
OwnerExternal	External owner status	External/Not External	*Bit9*
NotReady	Not ready to function	–	*Bit10*
SetpointLockActive	Setpoint change is denied		*Bit13*
InterlockTrip	Interlocked trip is active		*Bit15*
Bit items derived from ConfigurationWord.			
OwnerSelect	Owner of control module	Operator/Program	*Bit0*
InterlockBypass	Interlock bypass command	Bypassed/Normal	*Bit1*
InputSource	Input source selection	Operator/Field	*Bit2*
Rearm	Reset detected error	Active/Inactive	*Bit3*
CommandSetpoint	Operator setpoint	On/Off	*Bit4*
SimulationFailAc-tive	Feedback detected error in simulation mode	Enable/Disable	*Bit12*
Simulation	Simulation mode	–	*Bit13*
Maintenance	Maintenance mode		*Bit14*
OutOfService	Valve unavailable	Out Of Service/In Service	*Bit15*
IO Items			
StatusWord	Status word	–	VALVEGP_ST.STW
ConfigurationWord	Configuration word		VALVEGP_ST.CFGW
Calculated Items			
Equipstatus	Current status of the valve. For more details, refer to EquipStatus calculated variable, page 251.	–	GPL_ValveStatus
EquipState	Output information, For more details, refer to EquipState calculated variable, page 252.		GPL_ ValveEquipState
Owner	Current owner of the valve. For more details, refer to calculated variable, page 58.		GPL_ OwnerExternalLo-cal
Mode	Current mode of the valve. For more details, refer to calculated variable, page 59.		GPL_Mode
Abnormal	Shows abnormal condition of the valve. For more details, refer to abnormal conditions, page 252.		GPL_Abnormal
ActionRequired	Shows action required condition of the valve.		GPL_ ActionRequired

Item name	Description	Enumeration	Address
	For more details, refer to action required conditions, page 252.		
Optional Items			
Local panel			
Bit items derived from LPStatusWord.			
`LocalPanelMode`	Control from field	Enabled/Disabled	*Bit0*
`ZeroMode`	valve control disabled		*Bit1*
`ControlSystemMode`	Control from remote		*Bit2*
`LockedOwner`	Owner change disabled from faceplate	Locked/Unlocked	*Bit3*
`LockedSetpoint`	Setpoint change disabled from faceplate		*Bit4*
`LocalPanelEnabled`	Control from field	Enabled/Disabled	*Bit5*
`PushButtonEnabled`	Local panel push buttons		*Bit6*
`VirtualLocalPanelEnabled`	Local panel enable from control		*Bit7*
`LocalPanelSelected`	Control from field	Selected/Not Selected	*Bit8*
`ZeroSelected`	Control disabled		*Bit9*
`ControlSystemSelected`	Control from remote		*Bit10*
`PushButtonClose`	Close push button signal	On/Off	*Bit11*
`PushButtonOpen`	Open push button signal		*Bit12*
Bit items derived from LPConfigurationWord.			
`VirtualSwitch`	Local panel selection from faceplate	Local/Control System	*Bit0*
IO Items			
`LPStatusWord`	Local panel status word	–	`VALVELPGP_ST.STW`
`LPConfigurationWord`	Local panel configuration word		`VALVELPGP_ST.CFGW`
Failures	Refer to the topic describing the variable items of failures, page 61.		
Interlocks	Refer to the topic describing the variable items of interlocks, page 60.		
Maintenance	Refer to the topic describing the variable items of maintenance, page 59.		

EquipStatus Calculated Variable

The table describes the *EquipStatus* calculated variable:

Equipment Status	Conditions	Enumeration
–	`Open` x `Closed`	Unknown
Valve Closed	Not `Open` x `Closed`	Closed
Valve Open	`Open` x Not `Closed`	Open
Valve Transition	`OP` x Not `Open`	Opening

Equipment Status	Conditions	Enumeration
	Not OP x Not Closed	Closing

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	Conditions	Enumeration
Passive with Out of Service	OOS	Close
Passive	Not OOS x Fail x Not Maint	Failed
Active with Interlock	Not OOS x Not Fail x ILCK x Not Maint x OP	Open Interlocked
Passive with Interlock	Not OOS x Not Fail x ILCK x Not Maint x Not OP	Close Interlocked
Active	Not OOS x Not Fail x Not ILCK x Not Maint x OP	Open
Passive to Active	Not OOS x Not Fail x ILCK x OP x Not Maint x Not Open	Opening
Active	Not OOS x Not Fail x ILCK x OP x Not Maint x Open	Open
Passive with Out of Service	Not OOS x Not Fail x ILCK x Not OP x Maint	Close
Active to Passive	Not OOS x Not Fail x ILCK x OP x Not Maint x Not Closed	Closing
Passive	Not OOS x Not Fail x ILCK x OP x Not Maint x Closed	Close

Abnormal Conditions

The table describes the abnormal conditions of valve:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
AbnormalOwner	Abnormal owner status appears, if the current owner is not a normal owner which configured at the instance level.	*Bit0*
AbnormalMode	Abnormal mode status appears, if the current operating mode is not normal mode.	*Bit1*
AbnormalInterlockBy-pass	Abnormal interlock bypass condition appears, if the Interlock bypass is enabled.	*Bit2*
AbnormalOperatorTab	Abnormal status of operator tab. This condition appears, if the Interlock bypass is enabled.	*Bit8*
AbnormalFailTab	Abnormal status of **Failures** tab.	*Bit10*
AbnormalInterlockTab	Abnormal status of interlock tab.	*Bit11*
AbnormalEngineeringTab	Abnormal status of engineering tab.	*Bit12*

Action Required Conditions

The table describes the action required conditions of valve:

Item name	Description	Address
Bit items derived from ActionRequired variable word.		
ActionOperatorTab	Action required status of operator tab.	*Bit8*
ActionFailTab	Action required status of **Failures** tab.	*Bit10*
ActionInterlockTab	Action required status of interlock tab.	*Bit11*

Alarms

The table describes the advanced alarm items that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
`VlvMntManeuversNumber-sAlm`	Operations counter overflow	`GPL_BitValueCheck (StatusWord, 0)`
`VlvMntOperationHour-sAlm`	Running hours counter overflow	`GPL_BitValueCheck (StatusWord, 1)`
`Alarm`	Detection of unsuccessful operation	`GPL_BitValueCheck (StatusWord, 4)`
`Fail`	Inoperable valve	`GPL_BitValueCheck (StatusWord, 5)`
`ILCKTrip`	Interlock trip	`GPL_BitValueCheck (StatusWord, 15)`

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
`StateAlarm`	All changes in the valve status and output.
`ModeAlarm`	All changes in the mode of operation of the equipment.
`ActionAlarm`	All the operator actions in the supervision.

Trends

No trend item is used by Supervision components for `valve`.

Motorized Control Valve

What's in This Chapter

Template 254
Control 260
Optional Service - Local Panel (MVALVELPGP)..... 270
Supervision..... 274

Overview

This chapter describes the functionality of template, control services and Supervision functions of the motorized control valve.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$MValvedWithPosGP` template.

Description

General Description

The `$MValvedWithPosGP` is designed to control a motorized valve based on its required position (percentage of opening) and/ or its limits (open and close).

Function Description

The main functions of the motorized valve with position template are described in the following table:

Function	Description
Control	Controls the digital control signal of the element based on the configuration and the commands that the template receives according to the functions.
Owner selection	The template manages the control system level, which is the owner (operator or program). As a result, it is responsible for setting the set-point for the desired position.
Local and remote set-point	Enables to control the template with a set-point that is determined through a set-point selector (local or remote) in program owner. The local set-point is assigned from the sequential control while the remote set-point is assigned to the template control from the logic implemented in the continuous control.
Position detection	Allows to determine the actual position of the element to be controlled and monitored with the help of high and low limit switch.
Inoperable device	Allows to monitor inoperable device condition (optional). Template can be configured to withdraw output based on this condition.
Interlocking	The template gives a command to the device to move to the defined position in case of an active interlock. An interlock bypass function is available.
Detection of unsuccessful operation	If position is unsuccessful within a configured time after the output command is activated, template detects unsuccessful operation.
Manual resetting	Allows you to reset manually after the detection of an inoperable device condition or unsuccessful operation. After reset, device output will follow set point.

Optional Services

Available optional services are:

- **Interlocks**
- **Failures**
- **Local Panel**
- **Maintenance**

Parameters

Parameters

The `$MValvedWithPosGP` template defines operation of motorized valve without position. Alternative MValveWithPos facets enable the modification of the MValveWithPos that are available.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default value	Additional remarks
Configuration	**High present value**	`Float`	*100.0*	Maximum range of PV in engineering unit
	Low present value		*0.0*	Minimum range of PV in engineering unit
	Minimum position to consider valve open (%)		*95.0*	–
	Maximum position to consider valve closed (%)		*5.0*	–
Setpoint	**Reset on failure**	`Boolean`	*FALSE*	Resets the setpoint when the motor is non-operational.
	Track setpoint on interlock	`Boolean`	*TRUE*	The interlock setpoint is tracked by the equipment setpoint when the interlock condition is active.
	Force remote setpoint	`Boolean`	*FALSE*	Enables the remote control operation for motor.
	Force analog setpoint	`Boolean`	*FALSE*	Enables the analog control operation for motor.
Failure	**Motor confirmation timeout**	`Duration`	*00:00:30*	Maximum time for motor feedback confirmation.
	Valve confirmation timeout		*00:00:30*	Maximum time for valve feedback confirmation.
	Maximum limit switch time		*00:00:02*	Maximum time to lose limit switch same to be followed as `Mvalve`.
	Minimum alarm duration		*00:00:05*	Minimum time to maintain the confirmation alarm or detected error alarm.
	Disable alarm when motor is not running	`Boolean`	*FALSE*	Disables the alarm if device requires a reset after confirmation error. motor is stopped.
Rearm	**Required on failure**	`Boolean`	*TRUE*	Device requires a reset after device error or interlock trip alarm.
	Required on confirmation failure			Device requires a reset after confirmation error.
	Rearm resets conditions		*FALSE*	Rearms the Motor, interlock and detected error conditions.
Range	**Present value high value**	`Float`	*100.0*	High range in engineering units.

Section	Parameter description	Type	Default value	Additional remarks
	Present value low value		*0.0*	Low range in engineering units.

Inputs (Analog Input)

The table describes the analog inputs:

Section	Parameter	Type	Default value	Description
Configuration	**Include quality signal (Applicable for 'Int' data type)**	`Boolean`	*FALSE*	It provides signal quality informations like signal value, Error, COMSTATUS.
	Custom signal name	`String`	–	Instance name.
	Select analog channel data type	`Enum`	*Int (0)*	The data type of the signal value.

Inputs (Digital Input)

The table describes the digital input:

Name	Description
`OpenPosition`	It allows you to connect the open feedback of the valve.
`ClosePosition`	It allows you to connect the close feedback of the valve.
`ForwardRunning`	it allows to connect forward running feedback of the motor.
`ReverseRunning`	it allows to connect reverse running feedback of the motor.
`Fail`	It allows you to connect the detected error condition of the motor.
`ExternalControl`	It allows you to control the device externally.

For more details, refer to the topic describing the parameters of digital inputs, page 44.

Outputs (Digital Output)

The table describes the digital output:

Name	Description
`Open`	It allows you to connect the digital output signal of the motorized control valve to external device.
`Close`	It allows you to disconnect the digital output signal of the motorized control valve to external device.

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Failures

For more details, refer to the topic describing the parameters of failures, page 74.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Local Panel (Logic)

For more details, refer to the topic describing the parameters of logic, page 45.

Maintenance

For more details, refer to the topic describing the parameters of maintenance, page 89.

Local Panel (Digital Input)

The table describes the digital input:

Name	Description
LPMode	Local panel selection signal from mode selector switch.
ZeroMode	Zero/Off selection signal from mode selector switch.
CSMode	Control selection signal from selector switch.
OffPB	Motor stop signal from local panel.
OnPB	Motor on signal from local panel.
OpenPB	Motor start signal from local panel.
ClosePB	Motor stop signal from local panel.

For more details, refer to the topic describing the parameters of digital inputs, page 44.

Local Panel (Digital output)

The table describes the digital output:

Name	Description
PBLight	Output signal to indicate that push buttons are operational.

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Supervision

Supervision

The table describes the supervision parameters:

Section	Parameter	Type	Default value	Description
Configuration	**Enable Separate Alarms for Open and Close Confirmation Failure**	Boolean	*False*	It enables separate alarms for open and close confirmation error.
Operation	**Normal owner**	Enum	*Program and Cascade (5)*	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	**Set owner**	Enum	*Operator (1)*	Privilege to change owner. This parameter is used in Operator Tab, page 276.
	Set setpoint		*Operator (1)*	Privilege to change setpoint. This parameter is used in Operator Tab, page 276.
	Set mode - override		*Supervisor (2)*	Privilege to change override mode. This parameter is used in Engineering Tab, page 277.
	Set mode - simulation		*Engineer (5)*	Privilege to change simulation mode. This parameter is used in Engineering Tab, page 277.
	Set mode - maintenance		*Maintenance (3)*	Privilege to change maintenance mode. This parameter is used in Engineering Tab, page 277.
	Set mode - out of service		*Supervisor (2)*	Privilege to change out of service mode. This parameter is used in Engineering Tab, page 277.
	Set reset		*Operator (Confirmed) (10)*	Privilege to reset. This parameter is used in Operator Tab, page 276.
	Set alarm acknowledge		*Operator (1)*	Privilege to acknowledge alarm. This parameter is used in Alarm Tab, page 67.
	Set bypass conditions		*Supervisor (2)*	Privilege to set bypass conditions mode. This parameter is used in Engineering Tab, page 277.

Section	Parameter	Type	Default value	Description
	Set bypass all interlocks		*Supervisor (2)*	Privilege to bypass all interlocks. This parameter is used in Engineering Tab, page 277.
Range	**Present value high value**	`Float`	*100.0*	High range in engineering units
	Present value low value		*0.0*	Low range in engineering units
Trend Tag Disable	**Disable setpoint**	`Boolean`	*FALSE*	It disables setpoint trend tag.
Events	**Enable log**	`Boolean`	*FALSE*	Enable event logging for asset status. Example: Opening
Historize	**Hysteresis**	`Boolean`	*FALSE*	–
	Present value			–
	Setpoint			–
	Target setpoint			–
	Lower deviation threshold			–
	Upper deviation threshold			–

Asset and Display Parameters

For more details, refer to the topic describing the parameters of asset and display, page 43.

Alarm Parameters

For more details, refer to the topic describing the parameters of alarm, page 45.

Trend Parameters

For more details, refer to the topic describing the parameters of trend, page 47.

Failures

For more details, refer to the topic describing the parameters of failures, page 74.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Local Panel

For more details, refer to the topic describing the parameters of local panel, page 44.

Maintenance

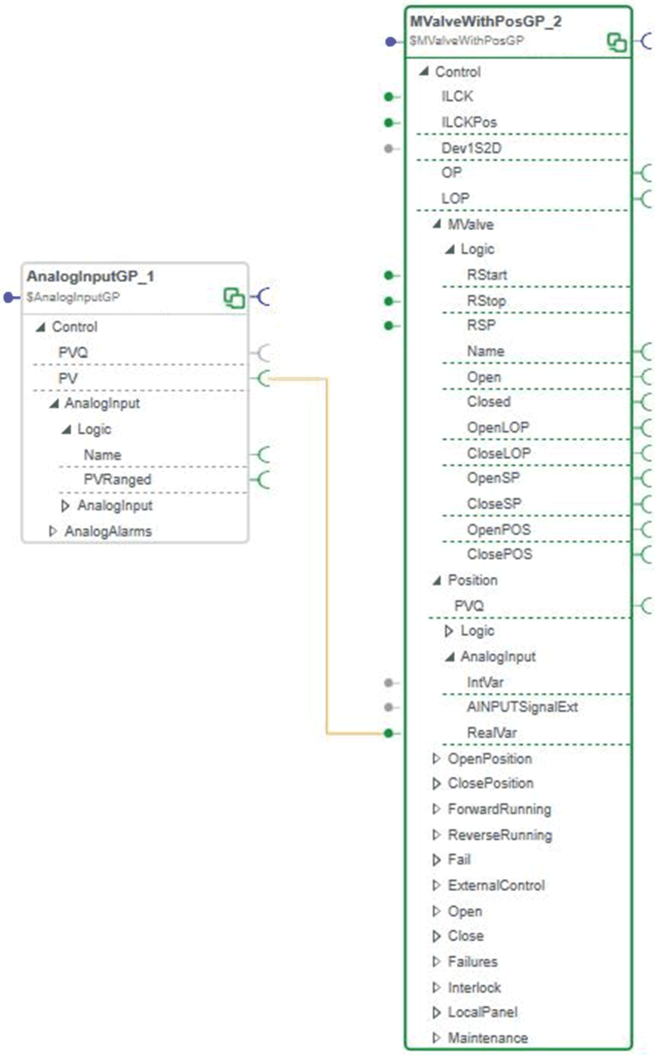
For more details, refer to the topic describing the parameters of maintenance, page 89.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

The following figure depicts an example of an instance of `$MValvedWithPosGP` (also applicable for `$MValveWithPosCE`) template as it appears in `Links Editor`.



The following table lists the link interface that creates link symbol on faceplate:

Interface name	Runtime parameter
`RSP`	*SetpointNavigate*
`PV`	*PVNavigate*
`PVRange`	
`PVRanged`	

For more details, refer to Supervision chapter > Faceplate > Navigation Link of General Purpose Library User Guide.

Refer to common control module template interface, page 47 for more details on common interfaces.

The table describes the control module template interfaces:

Interface identifier	Type/Role	Description	Example
`PV`	*$Real/Ref*	Analog Value output interface from Analog Input template.	Analog input value connected to Mvalve with positioner to control the position of its valve.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

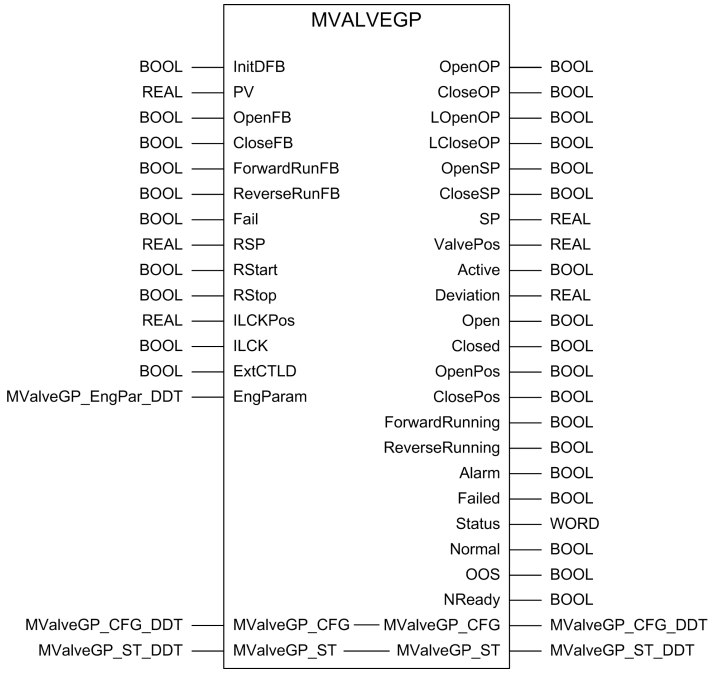
Overview

This chapter describes the `MVALVEGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

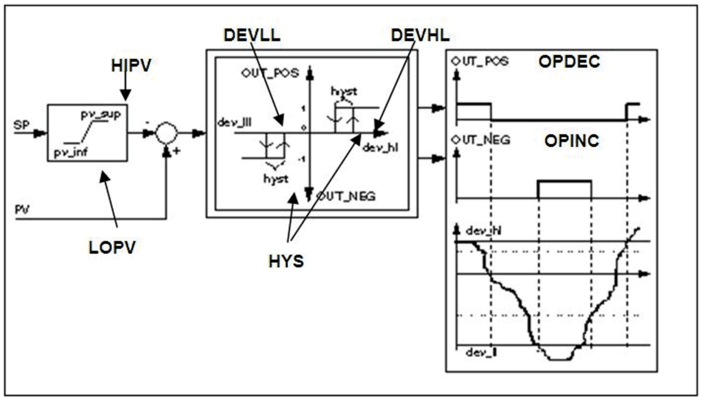
Input Parameter Description

Parameter	Type	Description
`InitDFB`	`BOOL`	When a rising edge occurs on this input, the internal timers of the DFB are reset and alarms are set to 0.
`PV`	`REAL`	Calculated measurement value usually in engineering units.
`OpenFB`	`BOOL`	1 = Indicates that the open valve position is reached. Used to connect to the digital input of the open-valve limit switch. Disabled internally by the DFB if no variable or value is connected.
`CloseFB`	`BOOL`	1 = indicates that the closed valve position is reached. Used to connect to the digital input of the closed-valve limit switch. Disabled internally by the DFB if no variable or value is connected.

Parameter	Type	Description
ForwardRunFB	BOOL	1 = Indicates to the DFB that motor is running in forward direction. Usually, it is used to connect the digital input of a motor for switching operation confirmation. Disabled internally by the DFB if no variable or value is connected.
ReverseRunFB	BOOL	1 = Indicates to the DFB that motor is running in reverse direction. Usually, it is used to connect the digital input of a motor for switching operation confirmation. Disabled internally by the DFB if no variable or value is connected.
Fail	BOOL	1 = Indicates to the DFB a detected error in the device.
RSP	REAL	Enables the remote position setpoint which is usually set by the continuous control and it has to be connected in program/remote.
RStart	BOOL	1 = Enables the operation of the *STEP3* block in automatic mode as described in the *Control Expert* document. 0 = It forces the OPINC and OPDEC outputs to 0, but the *STEP3* continues to run internally and to calculate the outputs. The valid **Start** command for the calculation is obtained from the following truth table. **Owner** / **Rem** / **Start is calculated as** OFF / OFF / SC.LStart OFF / ON / RStart ON / - / MValveGP_ST.LStart You can modify the set-point directly from the monitoring subsystem while this mode is active. If RStop is connected, then rising edge (Pulse based control) sends the command to start the motor. The value of pin is considered only when program/remote cascade mode is configured
RStop	BOOL	A rising edge sends a stop command only when program/remote cascade mode is configured.
ILCKPos	REAL	Position to be considered during interlock.
ILCK	BOOL	1 = Interlocks the device at the defined position. **NOTE:** If the device requires resetting (STW.ReamR = 1), the OP takes the value = 0 regardless of the ILCKPos value waiting for the operator to carry out the reset.
ExtCTLD	BOOL	1 = Indicates to the DFB whether the device is being controlled from outside of the system. Monitoring of confirmation alarms is stopped during external control. The detected error input and detected position error (OpenFB and CloseFB = 1) continues to be evaluated.

MValveGP_EngPar_DDT Type

Name	Type	Description
TimeOutValve	TIME	Maximum time for position confirmation of valve (0s = Disabled).
TimeOutMotor	TIME	Maximum time for feedback confirmation of motor (0s = Disabled).
ScanTime	TIME	Enables the time that the detected alarm signals are kept active to be configured. Enables the monitoring subsystem to acquire the data for detected alarms that are automatically reset.
ReactionTime	TIME	Enables to configure the time during which monitoring for the valve switching operations is enabled. During this time, a check is run to verify that the valve leaves the limit switch opposite to the operation that is carried out. When it has a value of 0, monitoring is disabled. This time is useful for detecting irregularity behavior in the valve and does not wait until the maximum stroke time is elapsed to detect them.
HighPV	REAL	Higher value of measurement range. This input limits the maximum set-point admitted by the controller. This input is internally connected to the *PARA.PV_SUP* parameter of the *STEP3* controller.
LowPV	REAL	Lower value of measurement range. This input limits the minimum set-point admitted by the controller. This input is internally connected to the *PARA.PV_INF* parameter of the *STEP3* controller.

Name	Type	Description
		
`HighThreshold`	`REAL`	Enables to configure the open-valve threshold for the analog position measurement as a percentage, that is, the threshold after which it is considered that the valve is completely open.
`LowThreshold`	`REAL`	Enables to configure the closed-valve threshold for the analog position measurement as a percentage, that is, the threshold after which it is considered that the valve is completely closed.
`ConfRearmEn`	`BOOL`	• 1 = A mandatory manual resetting is required when the detected error/interlock trip alarm condition disappears. • 0 = No manual resetting is required when the detected error/interlock Trip alarm condition disappears.
`FailRearmEn`	`BOOL`	• 1 = A manual resetting is required when the detected error/interlock trip alarm condition disappears. • 0 = No manual resetting is required when the detected error/interlock trip alarm condition disappears.
`DisStopAlarm`	`BYTE`	Disable device detected error alarm when motor is not running.
`ResetSPFail`	`BOOL`	Reset setpoint in case of detected error condition (1 = Restart is required if motor trips due to detected error and alarm).
`TrackSPEn`	`BOOL`	Track set-point in case of that the interlock condition is active. **NOTE:** The *TrackSPEn* parameter enables the interlock trip alarm when the device stopped due to interlock conditions.

Outputs

Output Parameter Description

Name	Type	Description
`OpenOP`	`BOOL`	1 = Field command signal for open position active. It always follows the `LOpenOP` command. **NOTE:** During simulation the signal will be deactivated.
`CloseOP`	`BOOL`	1 = Field command signal for close position active. It always follows the `LCloseOP` command. **NOTE:** During simulation the signal will be deactivated.
`LOpenOP`	`BOOL`	1 = Logical command signal for open position active. The current output (`OpenOP` output) is continuously copied to this variable except in the simulation mode. Refer to the `LOpenOP` and `LCloseOP` Calculation, page 264.
`LCloseOP`	`BOOL`	1 = Logical command signal for close position active. The current output (`CloseOP` output) is continuously copied to this variable except in the simulation mode. Refer to the `LOpenOP` and `LCloseOP` Calculation, page 264.
`OpenSP`	`BOOL`	Open setpoint Refer to the `SP` Calculation, page 264.
`CloseSP`	`BOOL`	Close setpoint

Name	Type	Description
		Refer to the `SP` Calculation, page 264.
`SP`	`REAL`	1 = Current set-point. For the output calculation that the DFB performs, refer to the `SP` Calculation, page 264.
`ValvePos`	`REAL`	Current position of the valve.
`Active`	`BOOL`	1 = Actuator is in `Run` mode.
`Deviation`	`REAL`	Deviation (*PV-SP*) extracted directly from the *STEP3* controller.
`Open`	`BOOL`	1 = Indicates the maximum opening position is reached. It is calculated based on the states obtained from the analog input and/or the digital input belonging to the **open-valve limit** switch. It is evaluated by considering other detectors. Refer to the output `Closed` and input/output MValveGP_ST.CFGW.Sim, page 266. Refer the `Open` and `Closed` calculation, page 265.
`Closed`	`BOOL`	1 = Indicates the closed-valve position is reached. It is calculated based on the states obtained from the analog input and/or digital input belonging to the **closed-valve limit** switch. It is evaluated by considering other detectors. Refer to the output `Open` and input/output MValveGP_ST.CFGW.Sim, page 266. Refer the `Open` and `Closed` calculation, page 265.
`OpenPos`	`BOOL`	1 = Indicates the maximum opening position is reached. It is calculated based on the states obtained from the analog input and/or the digital input belonging to the open valve limit switch. It is independent from the other detector if the `CloseFB` detector is available. Refer to the `OpenPos` and `ClosePos` calculation, page 265.
`ClosePos`	`BOOL`	1 = Indicates the closed valve position is reached. It is calculated based on the states obtained from the analog input and/or the digital input belonging to the close valve limit switch. It is independent from the other detector if the `OpenFB` detector is available. Refer to the OpenPos and ClosePos calculation, page 265.
`ForwardRunning`	`BOOL`	1 = The motor has reached the position corresponding to setpoint. It is evaluated by considering any detectors. Refer to the `ReverseRunning` output and MValveGP_ST.CFGW.Sim, page 266 input/output pin. Refer to the `ForwardRunning` and `ReverseRunning` feedback calculation, page 265.
`ReverseRunning`	`BOOL`	1 = The motor has reached the position corresponding to setpoint. It is evaluated by considering any detectors. Refer to the `ForwardRunning` output and MValveGP_ST.CFGW.Sim, page 266 input/output pin. Refer to the `ForwardRunning` and `ReverseRunning` feedback calculation, page 265.
`Alarm`	`BOOL`	1 = The detection of unsuccessful operation has occurred.
`Failed`	`BOOL`	1 = Irregularity in one of the auxiliary devices.
`Status`	`WORD`	States output by the *STEP3* controller on the *STATUS* output. For more details, see the *Control Expert* documentation for the *STEP3* controller. **Bit** / **Valor en dec.** / **Valor en hex.** / **Estado ENO** / **Significado** Bit 0 = 1 / 1 / 0x0001 / False / Error en un cálculo con valores de coma flotante Bit 1 = 1 / 2 / 0x0002 / False / Detección de un valor no permitido en una de las entradas de valores con coma flotante Bit 2 = 1 / 4 / 0x0004 / False / División por cero en un cálculo de valores con coma flotante Bit 3 = 1 / 8 / 0x0008 / False / Desborde de capacidad durante el cálculo de valores con coma flotante Bit 4 = 1 / 16 / 0x0010 / True / Se mostrarán los siguientes comportamientos: • `SP` se encuentra fuera de rango [`pv_inf, pv_sup`]: en este caso, `SP` se limita a `pv_inf` o `pv_sup`. • `dev_ll` > 0 o `dev_hl` < 0: el módulo utiliza el valor 0. • `hyst` se encuentra fuera del rango [0, mínimo (`dev_hl`, `-dev_ll`)]: el módulo utiliza un valor limitado a cero o al mínimo (`dev_hl`, `-dev_ll`).
`Normal`	`BOOL`	Indicates the valve is operating in normal mode.
`OOS`	`BOOL`	1 = Indicates the device is out of service.
`NReady`	`BOOL`	Indicates motor not ready for operation.

SP Calculation

The signal evaluation depending on the SC.Rem and RSP inputs and the MValveGP_ST.CFGW.Owner input/output is shown in the following table:

Owner	Rem	SP is calculated as
OFF	OFF	Sequential control (SC.LSP) and SC.LStart) or Pulse base control (SC.LSP, and SC.LRStart, SC.LRStop)
OFF	ON	Continous control (RSP) and RStart) or Pulse base control if RStop pin is connected (RSP and RStart, RStop)
ON	-	MValveGP_ST.LStartl and MValveGP_ST.SP

LOpenOP and LCloseOP Calculation

The evaluation of the device control depending on the interlock status and if the DFB requires resetting is shown in the following table:

STW.RearmR	SC.ILCKD	LOpenOP is calculated as:	LCloseOP is calculated as:
ON	-	OFF	OFF
OFF	ON	Refer to the below Interlock calculation table.	
OFF	OFF	OpenSP	CloseSP

Interlock Calculation

OpenOP and CloseOP in interlock state (output will be ON until feedback is received or any detected error triggers):

When OpenOP = OFF and CloseOP = ON					
OpenILCK	Open-ILCKPos	Close-ILCK	Close-ILCKPos	LOpenOP	LCloseOP
ON	OFF	OFF	OFF	Holds the last output state (Operator can change setpoint to Stop).	
ON	ON	OFF	OFF	ON	OFF
ON	OFF	ON	OFF	OFF	OFF
ON	OFF	ON	ON	OFF	ON
ON	ON	OFF	ON	ON	OFF
ON	ON	ON	ON	Holds the last output state.	
OFF	OFF	ON	OFF	OFF	OFF
OFF	OFF	ON	ON	OFF	ON
OFF	-	OFF	-	Holds the last output state.	

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

The *ConfRearmEn* engineering parameter has to be set to 1 for the *OP* pin of the function block output to become 0, under non-confirmed operations.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: Non-confirmed operations are those for which commands have been given and to which there is no confirming reply from the equipment.

Feedback Calculation for `Open`, `Closed`, `OpenPos`, `ClosePos`,

`Simulation`	`Open`	`Closed`	`OpenPos` is calculated as:	`ClosePos` is calculated as:
ON	`LOpenOP` and (`TimeOut`/2) or `PV` ≥ `HighThreshold`	`LCloseOP` and (`TimeOut`/2) or `PV` ≤`LowThreshold`	`OpenOP` and (`TimeOut`/2) or `PV` ≥ `HighThreshold`	`CloseOP` and (`TimeOut`/2) or `PV` ≤`LowThreshold`
OFF	`OpenFB` and (not `CloseFB` and not `ClosePos`)	`CloseFB` and (not `OpenFB` and not `OpenPos`)	`OpenFB` or `PV` ≥ `HighThreshold`	`CloseFB` or `PV` ≥ `LowThreshold`

Feedback Calculation for `ForwardRunning` and `ReverseRunning`

`ExtCTLD`	`ForwardRunningFB` connected	`ReverseRunningFB` connected	`Simulation`	`ForwardRunning` is calculated as:	`ReverseRunning` is calculated as:
-	-	-	ON	`LOpenOP` and (`TimeOut`/2)	`LCloseOP` and (`TimeOut`/2)
OFF	OFF	OFF	OFF	`LOpenOP`	`LCloseOP`
ON	OFF	OFF	OFF	OFF	OFF
OFF	OFF	ON	OFF	`ReverseRunningFB` and `OpenOP`	`ReverseRunningFB` and `CloseOP`
OFF	ON	OFF	OFF	`ForwardRunningFB` and `OpenOP`	`ForwardRunningFB` and `CloseOP`
OFF	ON	ON	OFF	`ForwardRunningFB` and (not `ReverseRunningFB`	`ReverseRunningFB` and (not `ForwardRunningFB`
ON	ON	OFF	OFF	OFF	OFF
ON	OFF	ON	OFF	OFF	OFF

Feedback Calculation for `ForwardRunningFB` and `ReverseRunningFB`

`ExtCTLD`	`ForwardRunningFB` connected	`ReverseRunningFB` connected	`Simulation`	`ForwardRunningFB` in `ST` is calculated as:	`ReverseRunningFB` in `ST` is calculated as:
-	-	-	ON	`LOpenOP` and (`TimeOut`/2)	`LCloseOP` and (`TimeOut`/2)
OFF	OFF	OFF	OFF	`LOpenOP`	`LCloseOP`
ON	OFF	OFF	OFF	OFF	OFF
OFF	OFF	ON	OFF	`ReverseRunningFB` and `OpenOP`	`ReverseRunningFB` and `CloseOP`
OFF	ON	OFF	OFF	`ForwardRunningFB` and `OpenOP`	`ForwardRunningFB` and `CloseOP`
OFF	ON	ON	OFF	`ForwardRunningFB`	`ReverseRunningFB`
ON	ON	OFF	OFF	OFF	OFF
ON	OFF	ON	OFF	OFF	OFF

Inputs/Outputs

Input/Output Parameter Description

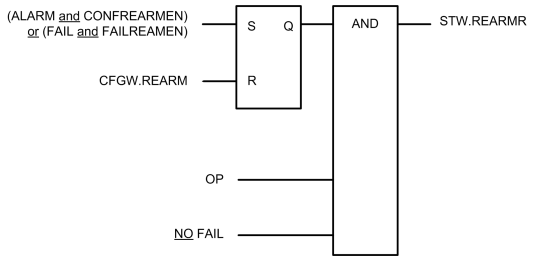
Parameter	Type	Description
MValveGP_CFG	MValveGP_CFG_DDT	Provides the data needed to configure the block from the monitoring subsystem or the sequential control.
MValveGP_ST	MValveGP_ST_DDT	Provides the data needed to monitor and/or control the block status.

MValveGP_CFG_DDT Type

Name	Type	Description
TargetSP	REAL	Read-only access This variable indicates the current set-point status.
DevLL	REAL	Enables to define the lower deviation threshold for the error detected (*PV-SP*). This value needs to be 0; otherwise, the DFB considers it 0. Refer to the operation graph, page 260 in *LowPV* and *STEP3* function block.
DevHL	REAL	Enables to define the upper deviation threshold for the error detected (*PV-SP*). This value needs to be ≥ 0; otherwise, the DFB considers it 0. Refer to the operation graph, page 260 in *LowPV* and *STEP3* function block.
Hysteresis	REAL	Defines the hysteresis value, which is considered by the *STEP3* algorithm when calculating the *OPINC* and *OPDEC* outputs. This value falls between 0 and the minimum value among *DevHL* and *DevLL* without considering the sign. The DFB helps to avoid an incorrect hysteresis inputs with the following rules: • If *CFG_Hysteresis* < 0 it is considered 0. • If *CFG_Hysteresis* > *DevHL* it is considered to be *DevHL*. • If *CFG_Hysteresis* >*DevLL* (unsigned) it is considered to be *DevLL* (unsigned).

MValveGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	Rem	*Bit0*	BOOL	• 1 = Enables the DFB to be configured to remote set-point Rem. • 0 = Enables the DFB to be configured to local set-point LSP.
	ILCK	*Bit2*	BOOL	1 = Interlocks the valve at the defined position. **NOTE:** If the device requires resetting (STW.Reamr = 1), the OP takes the value = 0 regardless of the ILCKPos value waiting for the operator to carry out the reset.
	LOpenOP	*Bit3*	BOOL	Refer to LOpenOP output pin, page 262.
	LCloseOP	*Bit4*	BOOL	Refer to LCloseOP output pin, page 262.
	ExtCTLD	*Bit5*	BOOL	Refer to ExtCTLD input pin, page 300.
	OpenPos	*Bit6*	BOOL	Refer to OpenPos output pin, page 262.
	ClosePos	*Bit7*	BOOL	Refer to ClosePos output pin, page 262.
	RearmR	*Bit8*	BOOL	Indicates if the valve requires resetting (1) after the detection of an inoperable device condition if it has been configured through the ConfRearmEn and/or FailRearmEn inputs.

Name	Bit Name	Bit	Type	Description
				The signal activation logic included in the DFB is as follows. 
	Failed	*Bit9*	BOOL	Refer to Failed output pin, page 262.
STW	Alarm	*Bit10*	BOOL	Refer to Alarm output pin, page 262.
	OpenAlarm	*Bit11*	BOOL	Refer to OpenAlarm, page 268.
	CloseAlarm	*Bit12*	BOOL	Refer to CloseAlarm, page 268.
	PosMismatch	*Bit14*	BOOL	Refer to PosMismatch, page 268.
	NReady	*Bit15*	BOOL	Refer to NReady output pin, page 262.
STW1	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	ForwardAlarm	*Bit0*	BOOL	Refer to ForwardAlarm, page 268.
	ReverseAlarm	*Bit1*	BOOL	Refer to ReverseAlarm, page 268.
	ForwardRunningFB	*Bit2*	BOOL	Refer to ForwardRunning output pin, page 262.
	ReverseRunningFB	*Bit3*	BOOL	Refer to ReverseRunning output pin, page 262.
	OpenSP	*Bit4*	BOOL	Refer to OpenSP output pin, page 262.
	CloseSP	*Bit5*	BOOL	Refer to CloseSP output pin, page 262.
	ILCKTrip	*Bit8*	BOOL	Refer to OpenILCK input pin, page 260.
	SPLocked	*Bit10*	BOOL	1 = Setpoint lock active.
	FailAlarm	*Bit11*	BOOL	1 = Detected error alarm is active.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read-write access to the data contained in this word.
	Owner	*Bit0*	BOOL	Read/write access Refer to Owner, page 268.
	ILCKBP	*Bit1*	BOOL	Read/write access 1 = Enables the interlock to be bypassed.
	LStart	*Bit2*	BOOL	Refer to LStart, page 268.
	Rearm	*Bit3*	BOOL	Write access Refer to Rearm, page 268.
	Sim	*Bit5*	BOOL	Refer to Sim, page 268.
	Maint	*Bit6*	BOOL	Refer to Maint, page 268.
	OOS	*Bit7*	BOOL	Refer to OOS output pin, page 262.
	SimFBFail	*Bit8*	BOOL	1 = Feedback will not simulate in simulation mode.
SP	–	–	REAL	Read/write access Refer to SP output pin, page 262.
ValvePos	–	–	REAL	Read-only access Refer to ValvePos output pin, page 262.

Public Variables

Public Variable Description

Variable	Type	Description
SC	MValveGP_SC_DDT	Provides the common data required to monitor and control the valve status from the sequential control.

MValveGP_SC_DDT Type

Name	Type	Description
Rem	BOOL	Read/write access 1 = Enables the DFB to be configured to remote set-point RSP 0 = Enables the DFB to be configured to local set-point LSP.
LSP	BOOL	Read/write access 1 = Enables the assignment of the target local set-point usually for the sequential control if the owner is the Program (Owner input/output is OFF) and the selected set-point is Local (SC.Rem public variable is OFF). 0 = The block automatically assigns the current value of the resulting set-point. It is described as follows:

Owner	Rem	SC.LSP
OFF (Program)	OFF	SC.LSP (last value or new value assigned from the sequential control)
OFF	ON	SP (calculated from the RSP)
ON (Operator)	ON	SP (last value or new value assigned by the Operator)

Name	Type	Description
TargetSP	BOOL	Read-only access This variable indicates the current set-point status before interlocking.
ValvePos	BOOL	Read-only access Current position of device (Read-only).
LStart	BOOL	Read/write access 1 = allows to enable*Start* command while the owner is the Program and Local; otherwise, the valid *Start* is copied to *SC.LStart*. Refer to the *RStart* input pin, page 260.
LRStart	BOOL	Read/write access Local start setpoint (Rising edge to start the actuator) (Read/Write).
LRStop	BOOL	Read/write access Local stop setpoint (Rising edge to stop the actuator) (Read/Write).
OpenOP	BOOL	Read-only access Refer to OpenOP output pin, page 262.
CloseOP	BOOL	Read-only access Refer to CloseOP output pin, page 262.
LOpenOP	BOOL	Read-only access Refer to LOpenOP output pin, page 262.
LCloseOP	BOOL	Read-only access Refer to LCloseOP output pin, page 262.

Name	Type	Description
OpenSP	BOOL	Read-only access Refer to OpenSP output pin, page 262.
CloseSP	BOOL	Read-only access Refer to CloseSP output pin, page 262.
Open	BOOL	Read-only access Refer to Open output pin, page 262.
Closed	BOOL	Read-only access Refer to Closed output pin, page 262.
OpenPos	BOOL	Read-only access Refer to OpenPos output pin, page 262.
ClosePos	BOOL	Read-only access Refer to ClosePos output pin, page 262.
ForwardRun-ningFB	BOOL	Read-only access Refer to ForwardRunning output pin, page 262.
ReverseRun-ningFB	BOOL	Read-only access Refer to ReverseRunning output pin, page 262.
OpenAlarm	BOOL	Read-only access 1 = Open position confirmation alarm active (Read-only).
CloseAlarm	BOOL	Read-only access 1 = Close position confirmation alarm active (Read-only).
ForwardAlarm	BOOL	Read-only access 1 = Motor forward confirmation alarm active (Read-only).
ReverseAlarm	BOOL	Read-only access 1 = Motor reverse confirmation alarm active (Read-only).
Failed	BOOL	Refer to Failed output pin, page 262.
PosMismatch	BOOL	Read-only access 1 = Indicates a position mismatch between digital and analog feedbacks.
ILCKD	BOOL	Read-only access The signal evaluation depending on the ILCK input.
Rearm	BOOL	Read-only access Allows the device to be reset (1). The block sets the signal to 0 after each execution and sets the signal indicating that the resetting is required (CFGW.Rearm) to 0.
RearmR	BOOL	Read/write access Refer to RearmR inputs/outputs pin, page 266.
Owner	REAL	Read-only access Enables to configure whether the set-point is set by the Program (0) or by the Operator (1).
Normal	BOOL	Read-only access Refer to Normal output pin, page 262.
Sim	BOOL	Read-only access Enables the simulation mode.
Maint	BOOL	Read-only access Maintenance mode is active.

Name	Type	Description
OOS	BOOL	Read-only access Refer to OOS output pin, page 262.
NReady	BOOL	Read-only access Refer to NReady output pin, page 262.
ExtCTLD	BOOL	Read-only access Refer to the ExtCTLD input pin, page 260.

Optional Service - Local Panel (MVALVELPGP)

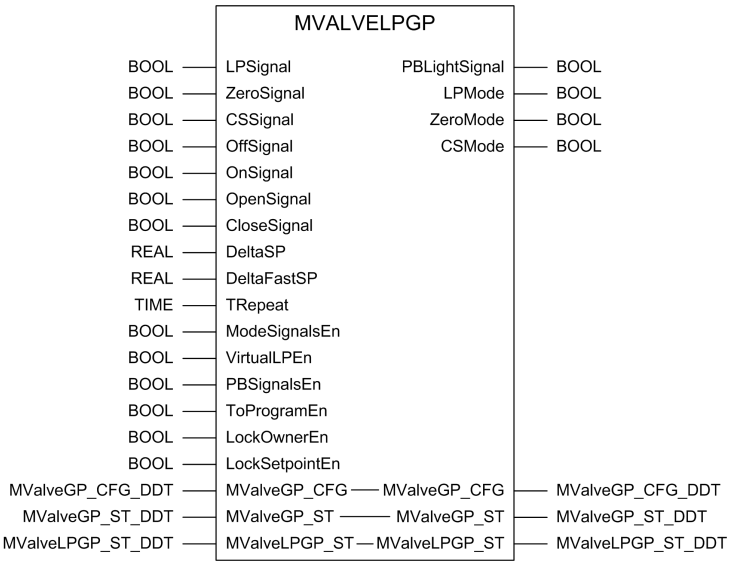
Overview

This chapter describes the optional services - local panel (MVALVELPGP) DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Name	Type	Description
LPSignal	BOOL	1 = The local panel mode switch is in the Local Panel position. Refer to the table included with the description of the LPMode output pin, page 272.
ZeroSignal	BOOL	1 = The local panel mode switch is in the zero position.This signal is optional even if a mode switch is available on the local panel. When this signal is a logical high, user is not allowed to operate the device from the faceplate and also from the DFB. Hence, all the functions in the faceplate will be disabled. Refer to the table included with the description of the LPMode output pin, page 272.

Name	Type	Description
CSSignal	BOOL	1 = The local panel mode switch is in the Control System position. Refer to the table included with the description of the LPMode output pin, page 272.
OffSignal	BOOL	1 = The OFF push button on the local panel is pressed.
OnSignal	BOOL	1 = The ON push button on the local panel is pressed.
OpenSignal	BOOL	1 = Indicates to the DFB the OPEN push button on the local panel is pressed.
CloseSignal	BOOL	1 = Indicates to the DFB the CLOSE push button on the local panel is pressed.
DeltaSP	REAL	Absolute set-point increase or decrease value when the OPEN or CLOSE push buttons are pressed.
DeltaFastSP	REAL	Absolute set-point increase or decrease value when the OPEN and CLOSE push buttons are being pressed.
TRepeat	TIME	Indicates the maximum time that needs to elapse before the system interprets that a command has been repeated when the push button is pressed. This time also determines when a DeltaFastSP increase or decrease has to be applied when the OPEN or CLOSE push button is pressed.
ModeSignalsEn	BOOL	1 = Enables the use of the Local Panel/Zero/ Control System (or Local Panel - Control System) mode switch on the local panel. Usually configured in development phase based on the characteristics of the local panel being used. If the mode switch is enabled, the LPSignal, ZeroSignal, and CSSignal inputs are considered to determine the operating modes of the local panel. Refer to the table included with the description of the LPMode output pin, page 272. **NOTE:** *ModeSignalsEn* input pin has higher priority than the *VirtualLPEn* input pin signal.
VirtualLPEn	BOOL	1 = This signal enables the operator to select the Local/Control System mode of operation from the HMI, so that the push button signals are enabled for operation. **NOTE:** When the owner is program, the Local/Control System mode selection drop-down list will be visible but disabled for operation however, when the owner is operator the Local Panel drop-down list will be accessible for operation. Refer to the table included with the description of the LPMode output pin, page 272.
PBSignalsEn	BOOL	1 = Enables to accept the OPEN and CLOSE push buttons from the local panel. This signal is not applicable if the mode switch is enabled (refer to the ModeSignalsEn input). This means that the push button signals are considered in the Local Panel mode. When the mode switch is disabled (ModeSignalsEn = 0), PBSignalsEn signal enables/disables the push button signals from control system itself based on relevant process conditions. SC.DisableLP public variable is also considered for determining whether the push buttons are enabled or not when *LPMode* pin signal is high as shown in the following table.

VirtualLPEn	ModeSignalsEn	PBSignal-sEn	SC. DisableLP	Push buttons enabled?
OFF	OFF	OFF	-	NO
OFF	OFF	-	ON	NO
OFF	OFF	ON	OFF	YES
OFF	ON	-	-	YES
ON	OFF	OFF	-	YES

Therefore, user can use this signal to enable/disable the push buttons on a simple local panel that does not feature a mode switch (Local Panel/Zero/Control System) based on the relevant process conditions.

ToProgramEn	BOOL	1 = Change to program mode when control is switched to control system mode and VirtualLPEn = 0.
LockOwnerEn	BOOL	1 = Disables the access of owner of the valve remaining as Operator while the local panel mode is Local Panel. The following table shows how the valve owner is evaluated based on the ToProgramEn and LockOwnerEn inputs and operating mode of the local panel:

Local Panel Mode	ToProgra-mEn	LockOwne-rEn	Owner
Zero	–	–	Operator
Switch to Local Panel	–	–	Operator
Local Panel	–	ON	Operator
Switch to Control System	ON	–	Program

Name	Type	Description
		In remaining cases, owner of the valve ceases to be defined from the `MVALVELPGP` DFB, and it enables to change it from the monitoring (HMI) system with the command word in the `MValveGP_ST.CFGW` input/output of the corresponding `MVALVEGP` DFB. In any case, consider that the owner of valve that had been set before switching to Local Panel mode is not memorized.
`LockSetpointEn`	`BOOL`	1 = Disables the setpoint of the valve device and lock it into the one defined on the local panel while the local panel mode is Local Panel and the device owner is the Operator. You can switch to the Program mode based on the signals that have been previously described even if the local panel mode is set to Local Panel. Therefore, user can activate (1) this signal to lock setpoint operation from the monitoring (HMI) system while the valve is controlled from the local panel.

Outputs

Output Parameter Description

Parameter	Type	Description
`PBLightSignal`	`BOOL`	1 = The push buttons on the local panel are fully operational, that is, the local panel is in Local Panel mode (either because the mode is selected with the corresponding switch or because the push buttons are enabled in the event that there is no selector switch available or the *VirtualLPEn* is high with the local panel mode enabled).
`LPMode`	`BOOL`	1 = The local panel mode is in Local Panel. The following table shows how the local panel operating mode is determined based on the `ModeSignalsEn`, `VirtualLPEn`, `PBSignalsEn`, `LPSignal`, `CSSignal` and `ZeroSignal`, input signals:

ModeSignalsEn	**VirtualLPEn**	**PBSignalsEn**	**LPSignal**	**CSSignal**	**ZeroSignal**	**OUTPUT**
0	0	0	0	0	0	CS mode
1	0	0	0	0	0	Zero mode
1	0	0	1	0	0	LP mode and PBLightSignal
1	0	0	0	1	0	CS mode
0	1	1	NA	NA	NA	CS mode
0	1	0	NA	NA	NA	CS mode
0	0	1	0	0	0	LP mode and PBLightSignal
1	1	0	1	0	0	LP mode and PBLightSignal
1	1	0	0	0	0	Zero mode
1	1	0	0	0	1	Zero mode
1	1	0	0	1	0	CS mode
1	0	1	0	1	0	CS mode
1	1	1	0	1	0	CS mode
1	0	0	0	0	1	Zero mode

`ZeroMode`	`BOOL`	1 = The local panel mode is in Zero. User is not allowed to operate the device from the faceplate and also from DFB. Hence, all the functions in the faceplate will be disabled. Refer to the table included with the description of the `LPMode` output.
`CSMode`	`BOOL`	1 = The local panel mode is in Control System. Refer to the table included with the description of the `LPMode` output.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
MVALVEGP_ST	MVALVEGP_ST_DDT, page 266	Data structure corresponding to the motorized valve is to be controlled from the local panel (for detailed information regarding this structure, refer to the MVALVEGP DFB, page 260. The structure provides the information needed for the operation of the MVALVELPGP DFB and enables this DFB to control the owner of the motorized valve.
MVALVEGP_CFG	MVALVEGP_CFG_DDT, page 266	Data structure corresponding to the motorized valve is to be controlled from the local panel (for detailed information regarding this structure, refer the MVALVEGP DFB). The structure provides the information needed for the operation of the MVALVELPGP DFB. It enables this DFB to control the set-point for the motorized valve.
MVal-veLPGP_ST	MValveLPGP_ST_DDT	Data structure that is used as an interface with the monitoring (HMI) system.

MVAVELPGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	LPMode	0	BOOL	Refer to the LPMode output pin, page 272.
	ZeroMode	1	BOOL	Refer to the ZeroMode output pin, page 272.
	CSMode	2	BOOL	Refer to the CSMode output pin, page 272.
	OwnerLocked	3	BOOL	Indicates whether the owner change buttons in the monitoring (HMI) system needs to be locked (1) or not (0). Is activated when: • The local panel mode is set to Zero or • It is set to Local Panel, and the LockOwnerEn input signal has been configured as active (1).
	SPLocked	4	BOOL	Indicates whether the set-point change and START buttons in the monitoring (HMI) system needs to be locked (1) or not (0). Is activated when: • The local panel mode is set to Zero or • It is set to Local Panel, and the LockSetpointEn input signal has been configured as active (1).
	ModeSignalsEn	5	BOOL	Refer to the ModeSignalsEn input pin, page 270.
	PBEnabled	6	BOOL	Indicates whether the push buttons of the local panel are enabled (1) or not (0). The signal is activated when at least one of the ModeSignalsEn and PBSignalsEn signal inputs is active.
	VirtualLPEn	7	BOOL	Refer to the VirtualLPEn input pin, page 270.
	LPSignal	8	BOOL	Refer to the LPSignal input pin, page 270.
	ZeroSignal	9	BOOL	Refer to the ZeroSignal input pin, page 270.
	CSSignal	10	BOOL	Refer to the CSSignal input pin, page 270.
	OffSignal	11	BOOL	Refer to the OffSignal input pin, page 270.
	OnSignal	12	BOOL	Refer to the OnSignal input pin, page 270.
	OpenSignal	13	BOOL	Refer to the OpenSignal input pin, page 270.
	CloseSignal	14	BOOL	Refer to the CloseSignal input pin, page 270.

Name	Bit Name	Bit	Type	Description
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
	LPEnabled	0	BOOL	Read/write access • 1 = Enables Local Panel mode. • 0 = Enables Control System mode.

Public Variables

Public Variable Description

Variable	Type	Description
SC	MVALVELPGP_SC_DDT	Provides the frequently needed data for monitoring and controlling the DFB.

MVALVELPGP_SC_DDT Type

NAME	Type	Description
LPMode	BOOL	Read-only access Refer to the LPMode output pin, page 272.
ZeroMode	BOOL	Read-only access Refer to the ZeroMode output pin, page 272.
CSMode	BOOL	Read-only access Refer to the CSMode output pin, page 272.
DisableLP	BOOL	Read/write access 1 = Disables the push buttons on the local panel. Only applicable when a local panel without mode switch configuration is used (ModeSignalsEn = 0).

Supervision

Overview

This section describes the Supervision functions and runtime services that are available for the motorized control valve.

Genies

Genie Properties

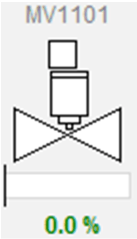
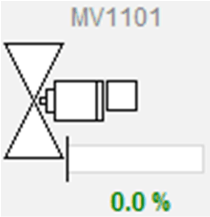
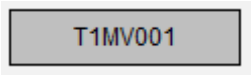
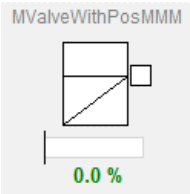
Refer to Genie Properties, page 62.

Genie Icons

Refer to Genie Icons, page 63.

Representation

The table describes the genies of the `gpl_valvemotorize` library:

Graphic symbol	Genie name	Description
	Motor2way_Horiz	Horizontal two-way valve. Horizontal bar with current value of the valve position.
	Motor2way_Vert	Vertical two-way valve. Horizontal bar with current value of the valve position.
	valvemtrzedinstan-ceid	Motorized valve instance.
	damper_motorize	Motorize valve - damper.

Faceplates

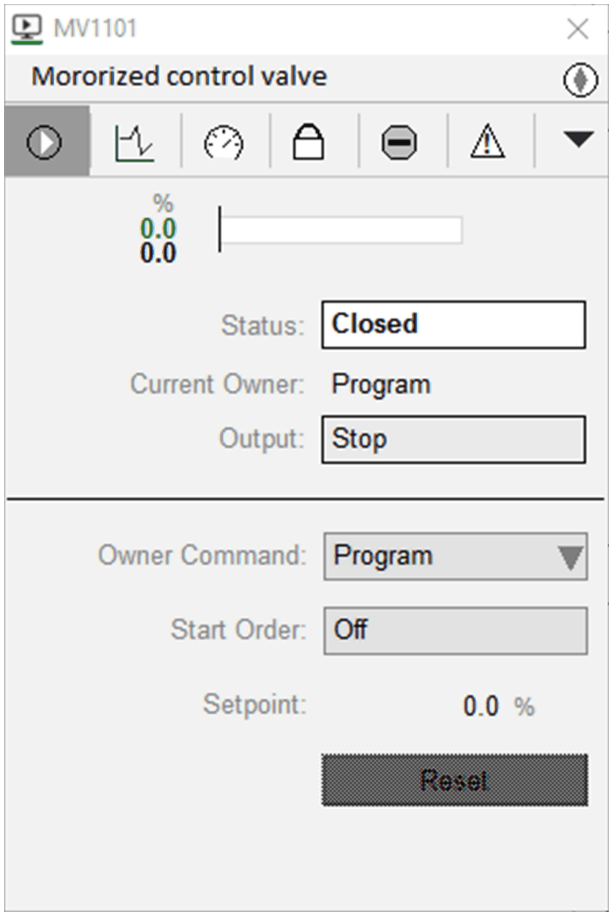
Overview

During operation, clicking a genie of the `Motorized control Valve` library allows you to display a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
  - Event, page 69
- Tabs for optional functions, which appear only if configured:
  - Interlock, page 87
  - Failure, page 78
  - Measures, page 70

OperatorTab

The figure shows an example of an operator tab:

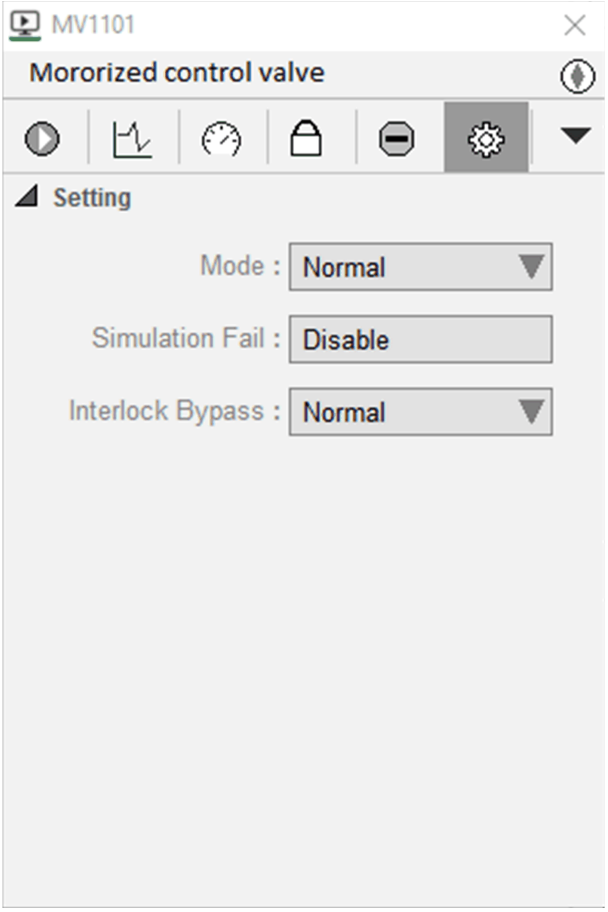


The table describes the control function of an operator tab:

Label	Item name	Description	Security parameter, page 141
Status	EquipStatus	Equipment status of motorized control valve. For more details, refer to EquipStatus calculated variable, page 280.	—
Current Owner	Owner	Current owner of motorized control valve. For more details, refer to operating owner, page 58.	
Output	EquipState	Output status of motorized control valve. For more details, refer to EquipState calculated variable, page 280.	
Owner Command	OwnerCommand	Owner command of motorized control valve (Program, Operator or Local Panel). **NOTE:** Local panel owner is enabled when the virtual panel is selected.	Set owner
Start Order	Start	Allows you to change the setpoint selection, if owner = operator.	Set setpoint
Setpoint	Setpoint	Allows you to change the setpoint, if owner = operator.	Set setpoint
–	Rearm	Allows you to reset manually after the detection of an inoperable device condition or unsuccessful operation. **NOTE:** When the control module is reset, the current setpoint shown in operator tab is effective. Also, when you click **Reset**, a dialog box opens to confirm.	Set rearm

Engineering Tab

The figure shows an example of an engineering tab:



The table describes the control function of engineering tab:

Label	Item name	Description	Security parameter, page 257
Setting			
Mode	Mode	Allows you to view and change the mode. For more details, refer to operating mode, page 59.	Set mode - normal, set mode - override, set mode - simulation, set mode - maintenance, and Set mode - out of service **NOTE:** Each configurable mode has the different security parameters
Simulation Fail	SimulationFailActive	Enable or disable feedback simulation. **NOTE:** The Simulation Fail will only be available when the parameter `SimulationFail` is created in the citect parameters. To create, go to **Setup** > **Parameters**. Enter the `GPL.Mode` in **Section Name**, `SimulationFail` in **Name** and set the **Value** = 1.	Set mode - simulation
Interlock Bypass	InterlockBypass	Interlock bypass status Allows you to bypass interlock.	Set bypass conditions, set bypass all interlocks
Maintenance Reset			

Label	Item name	Description	Security parameter, page 257
Reset Hours Count	MotMntResetHour-sCounter	Reset the hours counter.	Set reset
Reset Starts Count	MotMntResetManeuver-sCounter	Reset the operations counter.	Set reset

Items

Overview

This section describes the variables, equipstatus and equipstate variable, abnormal and action required conditions, alarms, events, and trends items of `motorized control valve`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`ProgramSetpointMode`	Setpoint source for program owner	Remote/Sequence	*Bit0*
`InterlockActive`	Interlock status	Interlocked/Healthy	*Bit2*
`SetpointMotor`	Motor operator setpoint	On/Off	*Bit3*
`DirectionSetpoint`	Direction setpoint	External/Not External	*Bit4*
`OwnerExternal`	External owner status	Interlocked/Healthy	*Bit5*
`FeedbackOpen`	Open feedback	Open/Not Open	*Bit6*
`FeedbackClosed`	Closed feedback	Closed/Not Closed	*Bit7*
`RequireRearm`	Require rearm	Required/Not Required	*Bit8*
`FailActive`	Valve inoperable	Active/Inactive	*Bit9*
`AlarmActive`	Feedback confirmation detected error	–	*Bit10*
`OpenAlarm`	Open alarm	–	*Bit11*
`CloseAlarm`	Close alarm	–	*Bit12*
`PositionMismatchAlarm`	Position mismatch error	Active/Inactive	*Bit14*
`NotReady`	Not ready to function	–	*Bit15*
Bit items derived from StatusWord1.			
`ForwardAlarm`	Forward alarm	–	*Bit0*
`ReverseAlarm`	Reverse alarm		*Bit1*
`ForwardRunningFB`	Forward running feedback		*Bit2*
`ReverseRunningFB`	Reverse running feedback		*Bit3*
`OpenSetpoint`	Open setpoint		*Bit4*
`CloseSetpoint`	Close setpoint		*Bit5*
`ILCKTripActive`	Interlock trip active		*Bit8*
`SetpointLockActive`	Setpoint lock active		*Bit10*
`FailAlarmActive`	Detected error alarm is active		*Bit11*
Bit items derived from ConfigurationWord.			
`OwnerSelect`	Owner of control module	Operator/Program	*Bit0*

Item name	Description	Enumeration	Address
InterlockBypass	Interlock bypass command	Bypassed/Normal	*Bit1*
Start	Motor start command	On/Off	*Bit2*
Rearm	Reset detected error	Active/Inactive	*Bit3*
Simulation	Simulation mode	–	*Bit5*
Maintenance	Maintenance mode		*Bit6*
OutOfService	valve unavailable	Out Of service/in service	*Bit7*
SimulationFailActive	Feedback detected error in simulation mode	Enable/Disable	*Bit8*
IO Items			
StatusWord	Status word	–	MVALVEGP_ST.STW
StatusWord1	Status word 1		MVALVEGP_ST.STW1
ConfigurationWord	Configuration word		MVALVEGP_ST.CFGW
Calculated Items			
Equipstatus	Current status of the motorized control valve with positioner For more details, refer to EquipStatus calculated variable, page 280.	–	GPL_ MotorizeValveSta-tus
EquipState	Output information For more details, refer to EquipState calculated variable, page 280.		GPL_ MotorizeValveE-quipState
Owner	Current owner of the motorized control valve with positioner For more details, refer to calculated variable, page 58.		GPL_ OwnerExternalLocal
Mode	Current mode of the motorized control valve with positioner For more details, refer to calculated variable, page 59.		GPL_ModeMValve
Abnormal	Shows abnormal condition of the motorized control valve with positioner For more details, refer to abnormal conditions, page 281.		GPL_Abnormal
ActionRequired	Shows action required condition of the motorized control valve with positioner For more details, refer to action required Conditions, page 281.		GPL_ActionRequired
Optional Items			
Local panel			
Bit items derived from LPStatusWord.			
LocalPanelMode	Control from field	Enabled/Disabled	*Bit0*
ZeroMode	motorized control valve with positioner control disabled		*Bit1*
ControlSystemMode	Control from remote		*Bit2*
LockedOwner	Owner change disabled from faceplate	Locked/Unlocked	*Bit3*
LockedSetpoint	Setpoint change disabled from faceplate		*Bit4*
LocalPanelEnabled	Control from field	Enabled/Disabled	*Bit5*

Item name	Description	Enumeration	Address
PushButtonEnabled	Local panel push buttons		*Bit6*
VirtualLocalPanelEnabled	Local panel enable from control		*Bit7*
LocalPanelSelected	Control from field	Selected/Not Selected	*Bit8*
ZeroSelected	Control disabled		*Bit9*
ControlSystemSelected	Control from remote		*Bit10*
PushButtonoff	Off push button signal	On/Off	*Bit11*
PushButtonOn	On push button signal		*Bit12*
PushButtonOpen	Open push button signal		*Bit13*
PushButtonClose	Close push button signal		*Bit14*
Bit items derived from LPConfigurationWord.			
VirtualSwitch	Local panel selection from faceplate	Local/Control System	*Bit0*
IO Items			
LPStatusWord	Local panel status word	–	MVALVELPGP_ST.STW
LPConfigurationWord	Local panel configuration word		MVALVELPGP_ST.CFGW
Failures	Refer to the topic describing the variable items of failures, page 61.		
Interlocks	Refer to the topic describing the variable items of interlocks, page 60.		
Maintenance	Refer to the topic describing the variable items of maintenance, page 59.		

EquipStatus Calculated Variable

The table describes the *EquipStatus* calculated variable:

Equipment Status	Conditions	Enumeration
–	(Open x Closed) + (FwdRun x RevRun)	Unknown
MValve Closed	Not Open x Closed	Closed
MValve Open	Not Open x Not Closed	Open
MValve Fully Open	Open x Not Closed	Fully Open

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	Conditions	Enumeration
Passive with Out of Service	OOS	Stop
Active in Maintenance	Not OOS x Maint x SPMotor x DirectionSP	Open
Passive with Out of Service	Not OOS x Maint x Not SPMotor x DirectionSP	Stop
Stop	Not OOS x Not Maint x Fail	Failed
Active1 with Interlock	Not OOS x Not Maint x Not Fail x ILCK x SPMotor x Not DirectionSP	Open Interlocked
Active2 with Interlock	Not OOS x Not Maint x Not Fail x ILCK x SPMotor x DirectionSP	Close Interlocked
Passive Interlock	Not OOS x Not Maint x Not Fail x ILCK x Not SPMotor x Not DirectionSP	Interlocked
Active	Not OOS x Not Maint x Not Fail x Not ILCK x SPMotor x Not DirectionSP	Open

Equipment State	Conditions	Enumeration
Passive to Active	Not `OOS` x Not `Maint` x Not `Fail` x Not `ILCK` x Not `SPMotor` x `DirectionSP`	Close
Passive	Not `OOS` x Not `Maint` x Not `Fail` x Not `ILCK` x Not `SPMotor` x Not `DirectionSP`	Stop

Abnormal Conditions

The table describes the abnormal conditions of motorized control valve:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
`AbnormalOwner`	Abnormal owner status appears if the current owner is not a normal owner which configured at the instance level.	*Bit0*
`AbnormalMode`	Abnormal mode status appears if the current operating mode is not normal mode.	*Bit1*
`AbnormalInterlockBypass`	Abnormal interlock bypass condition appears if the Interlock bypass is enabled.	*Bit2*
`AbnormalOperatorTab`	Abnormal status of operator tab. This condition appears if the Interlock bypass is enabled.	*Bit8*
`AbnormalFailTab`	Abnormal status of **Failures** tab.	*Bit10*
`AbnormalEngineeringTab`	Abnormal status of engineering tab.	*Bit12*

Action Required Conditions

The table describes the action required conditions of motorized control valve:

Item name	Description	Address
Bit items derived from ActionRequired variable word.		
`ActionOperatorTab`	Action required status of operator tab.	*Bit8*
`ActionFailTab`	Action required status of **Failures** tab.	*Bit10*
`ActionInterlockTab`	Action required status of interlock tab.	*Bit11*
`ActionDirection2In-terlockTab`	Action required status of abnormal direction 2 interlock tab.	*Bit14*

Alarms

The table describes the advanced alarm items that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
`Alarm`	Detection of unsuccessful operation	`GPL_BitValueCheck (StatusWord, 10)`
`AlarmValve`	Detection of unsuccessful operation	`GPL_BitValueCheck (StatusWord, 11,12)`
`PosMismatch`	Detection of unsuccessful operation	`GPL_BitValueCheck (StatusWord, 14)`
`AlarmMotor`	Inoperable motorized control valve	`GPL_BitValueCheck (StatusWord1, 0,1)`
`Fail`	Inoperable motorized control valve	`GPL_BitValueCheck (StatusWord1, 11)`

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
`StateAlarm`	All changes in the motorized control valve status and output.
`ModeAlarm`	All changes in the mode of operation of the equipment.
`ActionAlarm`	All the operator actions in the supervision.

trends

The table describes the trend items that are used by Supervision components:

Item Name	Expression	Description
`PV`	MVALVE_ST_ValvePos	Present value
`Setpoint`	MVALVE_ST_SP	Setpoint

Hand Valve

What's in This Chapter

Template 283
Control 285
Supervision 288

Overview

This chapter describes the functionality of template, control services and Supervision functions of the hand valve.

Template

Overview

This section describes functionality of the `$HandValveGP` template.

Description

General Description

The `$HandValveGP` process control module template is to manage valves without actuators and with one or two limit switches or any other type of positioner with two final positions.

The process template allows managing the associated devices from the sequential control, the continuous control, and/or the supervision system. It depends on their configuration and the system needs.

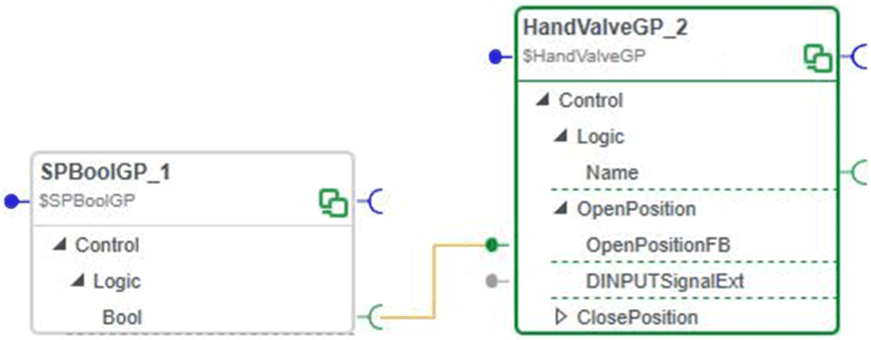
Function Description

The main functions of the hand valve template are described in the following table:

Function	Description
Position Detection	Allows to determine the actual position of the element to be controlled and monitored with the help of open and close feedback.
Mode selection	The object can be switched to different modes of operation like Normal, Override and Maintenance.
Position detection	Allows to determine the actual position of the element to be controlled and monitored with the help of open and close feedback.

Interfaces

This is an example of an instance of `$HandValveGP` (also applicable for `$HandValveCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



For interface identifiers, refer to the control module template interfaces, page 49.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Parameters

Parameters

The `$HandValveGP` process control module template is used to condition a digital signal usually associated with a physical output.

Control

Inputs (Digital Input)

The table describes the digital input:

Name	Description
`OpenPosition`	It allows you to connect the open feedback of the valve.
`ClosePosition`	It allows you to connect the close feedback of the valve.

For more details, refer to the topic describing the parameters of digital inputs, page 43.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Description
Security	Set mode - override	`Enum`	Supervisor (2)	Privilege to change override mode. This parameter is used in Engineering Tab, page 291.
	Set mode - maintenance		Maintenance (3)	Privilege to change maintenance mode. This parameter is used in Engineering Tab, page 291.
	Set alarm acknowledge		Operator (1)	Privilege to acknowledge alarm. This parameter is used in Alarm Tab, page 67.

Asset and Display Parameters

For more details, refer to the topic describing the parameters of asset and display, page 43.

Data

The table describes the data parameters:

Section	Parameter description	Type	Default	Description
Events	Enable log	`Boolean`	False	Enable event logging for asset status. Example: Opening

Asset and Display Parameters

For more details, refer to the topic describing the parameters of asset and display, page 43.

Alarm Parameters

For more details, refer to the topic describing the parameters of alarm, page 45.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Control

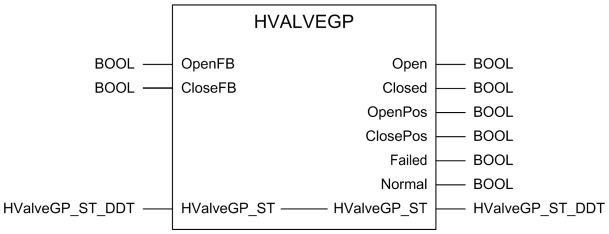
Overview

This chapter describes the `HVALVEGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Name	Type	Description
OpenFB	BOOL	1 = Indicates to the DFB that high limit position is reached. Usually, it is used to connect the digital input of a positioner high limit switch. The input pin is considered, if an input parameter is connected or linked.
CloseFB	BOOL	1 = Indicates to the DFB that low limit position is reached. Usually, it is used to connect the digital input of a positioner low limit switch. The input pin is considered, if an input parameter is connected or linked.

Outputs

Output Parameter Description

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>Open</td><td>BOOL</td><td>1 = Determines the device has reached the open-valve position considering the status of the connected detectors and/or the simulated status. Refer to the following table.</td></tr>
<tr><td>Closed</td><td>BOOL</td><td>1 = Determines the device has reached the closed-valve position considering the status of the connected detectors and/or the simulated status.

Refer to the following table:
<table>
<tr><th>OpenFB connected</th><th>CloseFB connected</th><th>Override</th><th>Open is calculated as:</th><th>Close is calculated as:</th></tr>
<tr><td>–</td><td>–</td><td>–</td><td>OverridePos</td><td>not Override-Pos</td></tr>
<tr><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td></tr>
<tr><td>ON</td><td>OFF</td><td>OFF</td><td>OpenFB</td><td>not OpenFB</td></tr>
<tr><td>OFF</td><td>ON</td><td>OFF</td><td>Not CloseFB</td><td>CloseFB</td></tr>
<tr><td>ON</td><td>ON</td><td>OFF</td><td>OpenFB and not CloseFB</td><td>CloseFB and not OpenFB</td></tr>
</table></td></tr>
<tr><td>OpenPos</td><td>BOOL</td><td>1 = Indicates the high position has been reached based on the detector connected on OpenFB.</td></tr>
<tr><td>ClosePos</td><td>BOOL</td><td>1 = Indicates the low position has been reached based on the detector connected on CloseFB.</td></tr>
<tr><td>Failed</td><td>BOOL</td><td>1 = Indicates an inoperable device when the OpenFB and CloseFB inputs are active at the same time.</td></tr>
<tr><td>Normal</td><td>BOOL</td><td>1 = Normal mode active.</td></tr>
</table>

Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
HValveGP_ST	HValveGP_ST_DDT	Provides the data necessary to monitor the DFB status.

HValveGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	OpenPos	*Bit0*	BOOL	Refer to the OpenPos output pin, page 286.
	ClosePos	*Bit1*	BOOL	Refer to the ClosePos output pin, page 286.
	Failed	*Bit2*	BOOL	Refer to the Failed output pin, page 286.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
	Override	*Bit0*	BOOL	1 = Override mode active.
	OverridePos	*Bit1*	BOOL	1 = Override input position (1 = Open, 0 = Close).
	Maint	*Bit14*	BOOL	1 = Maintenance mode active.

Public Variables

Public Variable Description

Variable	Type	Description
SC	HValveGP_SC_DDT	Provides the frequently needed data to monitor the DFB status.

HValveGP_SC_DDT Type

Name	Type	Description
Open	BOOL	Read-only access Refer to the Open output pin, page 286.
Closed	BOOL	Read-only access Refer to the Closed output pin, page 286.
OpenPos	BOOL	Read-only access Refer to the OpenPos output pin, page 286.
ClosePos	BOOL	Read-only access Refer to the ClosePos output pin, page 286.
Failed	BOOL	Read-only access Refer to the Failed output pin, page 286.
Normal	BOOL	Read-only access Refer to the Normal output pin, page 286.
Override	BOOL	Read/write access Rising edge to activate override mode (1 = Override mode active) (Read/Write).
Maint	BOOL	Read-only access 1 = Maintenance mode active.

Supervision

Overview

This section describes the Supervision functions and runtime services that are available for the hand valve.

Genies

Genie Properties

Refer to Genie Properties, page 62

Genie Icons

Refer to Genie Icons, page 63

Representation

The table describes the genies of the `gpl_valvehand` library:

Graphic Symbol	Genie Name	Description
	2way_Horiz	Horizontal two-way valve.
	2way_Vert	Vertical two-way valve.
	3way_Horiz_ Downleft	Horizontal three-way valve (down to left way when closed, down to right way when open).
	3way_Horiz_ Downright	Horizontal three-way valve (down to right way when closed, down to left way when open).
	3way_Horiz_ Leftdown	Horizontal three-way valve (left to down way when closed, left to right way when open).
	3way_Horiz_ Leftright	Horizontal three-way valve (left to right way when closed, left to down way when open).
	3way_Horiz_ Rightdown	Horizontal three-way valve (right to down way when closed, right to left way when open).
	3way_Horiz_ Rightleft	Horizontal three-way valve right to left way when closed, right to down way when open).

Graphic Symbol	Genie Name	Description
	3way_Vert_ Downright	Vertical three-way valve (down to right way when closed, down to up way when open).
	3way_Vert_Downup	Vertical three-way valve (down to up way when closed, down to right way when open).
	3way_Vert_ Rightdown	Vertical three-way valve (right to down way when closed, right to up way when open).
	3way_Vert_Rightup	Vertical three-way valve (right to up way when closed, right to down way when open).
	3way_Vert_Updown	Vertical three-way valve (up to down way when closed, up to right way when open).
	3way_Vert_Upright	Vertical three-way valve (up to right way when closed, up to down way when open).

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

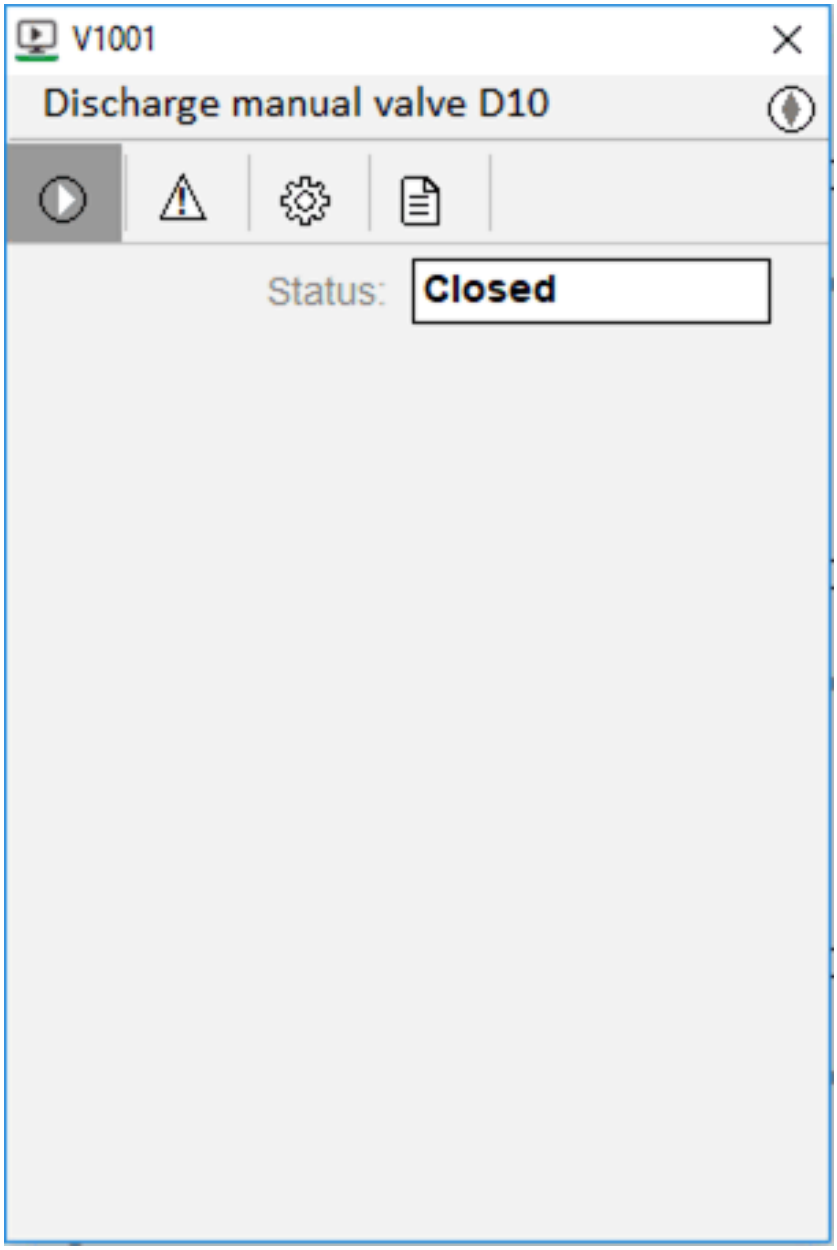
Available Tabs

During operation, clicking an motor genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
  - Event, page 69

Operator Tab

The figure shows an operator tab:

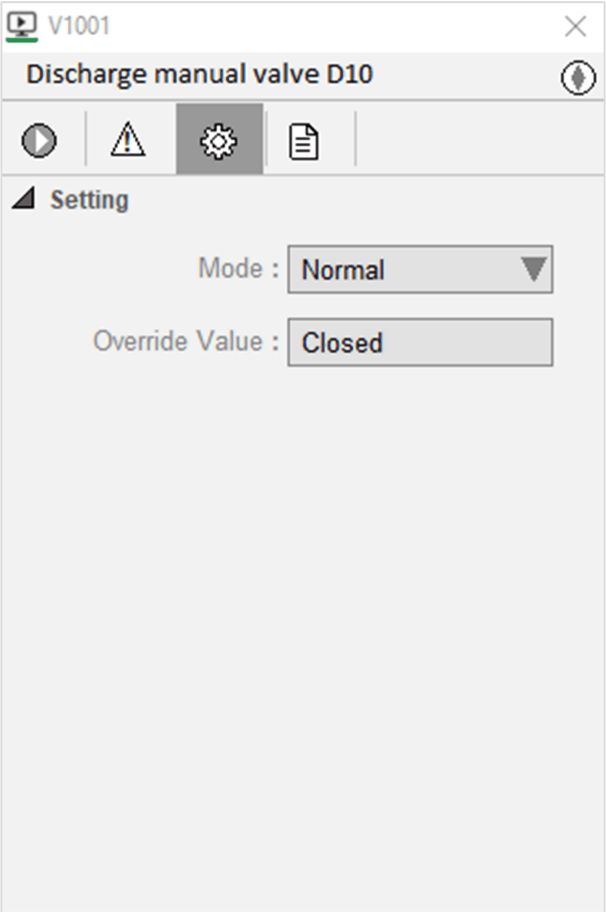


The table describes the control function of an operator tab:

Label	Item Name	Description	Security parameter, page 284
Status	EquipStatus	Equipment status of hand valve. For more details, refer to EquipStatus calculated variable, page 292.	—

Engineering Tab

The figure shows an engineering tab:



The table describes the control function of an engineering tab:

Label	Item Name	Description	Security parameter, page 284
Mode	Mode	Allows you to view and change the mode. For more details, refer to operating mode, page 59.	Set mode - override, and set mode - maintenance **NOTE:** Each configurable mode has the different security parameters
Override Value	OverrideValue	Override position value	Set mode - override

Items

Overview

This section describes the variables, equipstatus variable, abnormal and action required conditions, alarms, events, and trends items `of hand valve`.

Variables

The table describes the variable items that are used by Supervision components:

Item Name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`FeedbackOpen`	Open feedback	Open/Not Open	*Bit0*

Item Name	Description	Enumeration	Address
`FeedbackClosed`	Closed feedback	Closed/Not Closed	*Bit1*
`FailActive`	Valve inoperable	Active/Inactive	*Bit2*
Bit items derived from ConfigurationWord.			
`InputSource`	Input source selection	Operator/Field	*Bit0*
`OverrideValue`	Override position value	Open/Closed	*Bit1*
`Maintenance`	Maintenance mode	–	*Bit14*
IO Items			
`StatusWord`	Status word	–	`HVALVEGP_ST.STW`
`ConfigurationWord`	Configuration word		`HVALVEGP_ST.CFGW`
Calculated Items			
`Equipstatus`	Current status of the hand valve For, more details, refer to EquipStatus calculated variable, page 292.	–	`GPL_ HandValveStatus`
`Mode`	Current mode of the hand valve For, more details, refer to calculated variable, page 59.		`GPL_ ModeSignalInput`
`Abnormal`	Shows abnormal condition of the hand valve For, more details, refer to abnormal conditions, page 292.		`GPL_ HandValveAbnormal`
`ActionRequired`	Shows action required condition of the hand valve For more details, refer to action required conditions, page 293.		`GPL_ ActionRequired`

EquipStatus Calculated Variable

The table describes the *EquipStatus* calculated variable:

Equipment Status	Conditions	Enumeration
–	`Open` x `Closed`	Unknown
HValve Closed	`Open` x Not `Closed`	Open
HValve Open	Not `Open` x `Closed`	Closed
HValve Transition	Not `Open` x Not `Closed`	Opening

Abnormal Conditions

The table describes the abnormal conditions of hand valve:

Item Name	Description	Address
Bit items derived from Abnormal variable word.		
`AbnormalMode`	Abnormal mode status appears, if the current operating mode is not normal mode.	*Bit0*
`AbnormalOperatorTab`	Abnormal status of operator tab. This condition appears, if the Interlock bypass is enabled.	*Bit8*
`AbnormalEngineeringTab`	Abnormal status of engineering tab.	*Bit12*

Action Required Conditions

The table describes the action required conditions of hand valve:

Item Name	Description	Address
Bit items derived from ActionRequired variable word.		
`ActionOperatorTab`	Action required status of operator tab.	*Bit8*

Alarms

The table describes the advanced alarm items that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
`Fail`	Inoperable hand valve	`GPL_BitValueCheck (StatusWord, 2)`

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item Name	Description
`StatusAlarm`	All changes in the hand valve status and output.
`ModeAlarm`	All changes in the mode and owner.
`ActionAlarm`	All actions by the operator.

Trends

No trend item is used by Supervision components for `hand valve`.

Dual Output Valve

What's in This Chapter

Template 294
Control 299
Optional Service - Local Panel (DVALVELPGP) 309
Supervision 313

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the dual output valve.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$DualOPValveGP` template.

Description

General Description

The `$DualOPValveGP` process control module template is designed to read signals, and optionally, signal quality data from analog input modules. Alternatively, it can receive signals from an external variable of type Int or Real.

Function Description

The main functions of the template are described in the following table:

Function	Description
Digital Input	The template can generate two outputs based on one logical input.
Operating modes	The template provides continuous or pulse type digital output based on the digital input, limit switches and selected mode. This control function provides five different modes of operation: • `MODE` 3: Till position is confirmed (release the start/open or stop/close command once the position is reached). ◦ When input `OP` is set to true, `OPH` is set to true until the high position (`ZSHPOS`) is reached. ◦ When input `OP` is set to false, `OPL` is set to true until the low position (`ZSLPOS`) is reached. ◦ After initialization, outputs (`OPH` or `OPL`) are triggered based on the input signal and position feedback signals. • `MODE` 4: Till position is confirmed + extra time (release the start/open or stop/close command once the position is reached + extra pulse time to stroke the output) ◦ When input `OP` is set to true, `OPH` is set to true until the high position (`ZSHPOS`) is reached + defined pulse time duration (`OPHPulsetime`). ◦ When input `OP` is set to false, `OPL` is set to true until the low position is reached (`ZSLPOS`) + defined pulse time duration (`OPLPulsetime`). ◦ After initialization, outputs (`OPH` or `OPL`) are triggered based on the input signal and position feedback signals. **NOTE:** For invalid entry of `MODE` (for example, `MODE` < 0 or `MODE` > 4) the DFB defaults to `MODE` 0.

Optional Services

Available optional services are:

- **Failures**
- **Interlocks**
- **Local Panel**
- **Maintenance**

Parameters

Parameters

The `$DualOPValveGP` templates provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default value	Additional remarks
Setpoint	**Reset on failure**	`Boolean`	*FALSE*	Resets the setpoint when the motor is inoperable.
	Track setpoint on interlock	`Boolean`	*TRUE*	The interlock setpoint is tracked by the equipment setpoint when the interlock condition is active.
	Force remote setpoint	`Boolean`	*FALSE*	Enables the remote control operation for motor.
Failure	**Confirmation timeout**	`Duration`	*00:00:03*	Maximum time for feedback confirmation.

Section	Parameter description	Type	Default value	Additional remarks
	Minimum alarm duration		*00:00:05*	Minimum time to maintain the confirmation alarm or detected error alarm.
Rearm	**Required on failures**	`Boolean`	*TRUE*	Device requires a reset after device error or interlock trip alarm.
	Required on confirmation failure		*FALSE*	Device requires a reset after confirmation error.
	Rearm resets conditions			Rearms the valve, interlock and detected error conditions.
Configuration	**Mode of operation**	`Enum`	*Mode 0 - Continuous output (0)*	–
Time	**Time delay between switching of outputs**	`Duration`	*00:00:00*	–
	OPH pulse time duration	`Duration`	*00:00:03*	–
	OPL pulse time duration	`Duration`	*00:00:03*	–
	Maximum time to lose limit switch (0s = Disabled)	`Duration`	*00:00:02*	–

Inputs (Digital Input)

The table describes the digital input:

Name	Description
`OpenPosition`	It allows to connect the open feedback of the valve.
`ClosePosition`	It allows to connect the closed feedback of the valve.
`Fail`	It allows to connect the detected error conditions of the valve.
`ExternalControl`	It allows to control the device externally.
`RSPSel`	It allows to connect the remote set-point select condition of the valve.

For more details, refer to the topic describing the parameters of digital inputs, page 44.

Outputs (Digital Output)

The table describes the digital output:

Name	Description
`Open`	It allows to connect the digital output signal of the valve to external device.
`Close`	It allows to disconnect the digital output signal of the valve to external device.

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Failures

For more details, refer to the topic describing the parameters of failures, page 74.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Local Panel (Logic)

For more details, refer to the topic describing the parameters of logic, page 45.

Maintenance

For more details, refer to the topic describing the parameters of maintenance, page 89.

Local Panel (Digital Input)

The table describes the digital input:

Name	Description
LPMode	Local panel selection signal from mode selector switch.
ZeroMode	Zero/Off selection signal from mode selector switch.
CSMode	Control selection signal from selector switch.
OFF	Motor start signal from local panel.
ON	Motor stop signal from local panel.

For more details, refer to the topic describing the parameters of digital inputs, page 44.

Local Panel (Digital output)

The table describes the digital output:

Name	Description
PBLight	Output signal to indicate that push buttons are operational.

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default value	Additional remarks
Configu-ration	**Enable Separate Alarms for Open and Close Confirmation Failure**	Boolean	*FALSE*	It enables separate alarms for open and close confirmation error.
Operation	**Normal owner**	Enum	*Program and Cascade (5)*	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	**Set owner**	Enum	*Operator (1)*	Privilege to change owner. This parameter is used in Operator Tab, page 316.
	Set setpoint		*Operator (1)*	Privilege to change setpoint. This parameter is used in Operator Tab, page 316.
	Set mode - override		*Supervisor (2)*	Privilege to change override mode. This parameter is used in Engineering Tab, page 317.
	Set mode - simulation		*Engineer (5)*	Privilege to change simulation mode. This parameter is used in Engineering Tab, page 317.
	Set mode - maintenance		*Maintenance (3)*	Privilege to change maintenance mode. This parameter is used in Engineering Tab, page 317.

Section	Parameter description	Type	Default value	Additional remarks
	Set mode - out of service		*Supervisor (2)*	Privilege to change out of service mode. This parameter is used in Engineering Tab, page 317.
	Set reset		*Operator (Confirmed) (10)*	Privilege to reset. This parameter is used in Operator Tab, page 316.
	Set parameters		*Engineer (5)*	Privilege to set parameters. This parameter is used in Engineering Tab, page 317.
	Set alarm acknowledge		*Operator (1)*	Privilege to acknowledge alarm. This parameter is used in Alarm Tab, page 67.
	Set bypass conditions		*Supervisor (2)*	Privilege to set bypass conditions mode. This parameter is used in Engineering Tab, page 317.
	Set bypass all interlocks		*Supervisor (2)*	Privilege to bypass all interlocks. This parameter is used in Engineering Tab, page 317.
	Set asset maintenance		*Operator (1)*	Privilege to set maintenance mode. This parameter is used in Engineering Tab, page 317.

Data

The table describes the data parameters:

Section	Parameter description	Type	Default value	Additional remarks
Events	**Enable log**	`Boolean`	*FALSE*	Enable event logging for asset status.
Historize	**Setpoints**	`Boolean`	*FALSE*	It allows to historize the change in setpoint..
	Owner/Mode			It allows to historize the change in mode and owner.

Asset and Display Parameters

For more details, refer to the topic describing the parameters of asset and display, page 43.

Alarm Parameters

For more details, refer to the topic describing the parameters of alarm, page 45.

Failures

For more details, refer to the topic describing the parameters of failures, page 74.

Local Panel

For more details, refer to the topic describing the parameters of local panel, page 44.

Maintenance

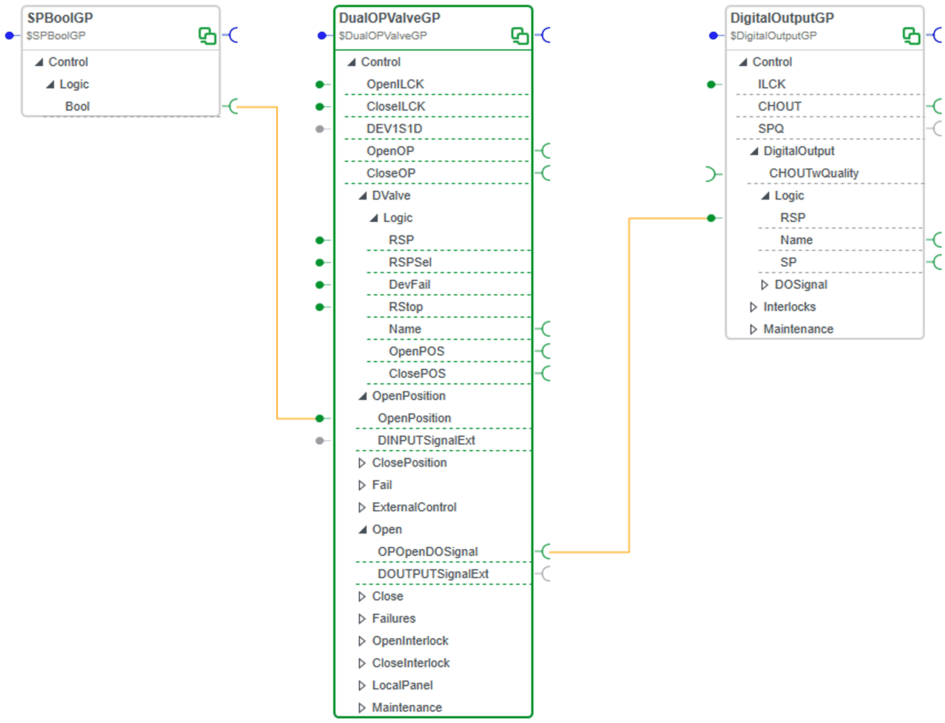
For more details, refer to the topic describing the parameters of maintenance, page 89.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

The following figure depicts the `$DualOPValveGP` (also applicable for `$DualOPValveCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



For interface identifiers, refer to the control module template interfaces, page 49.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

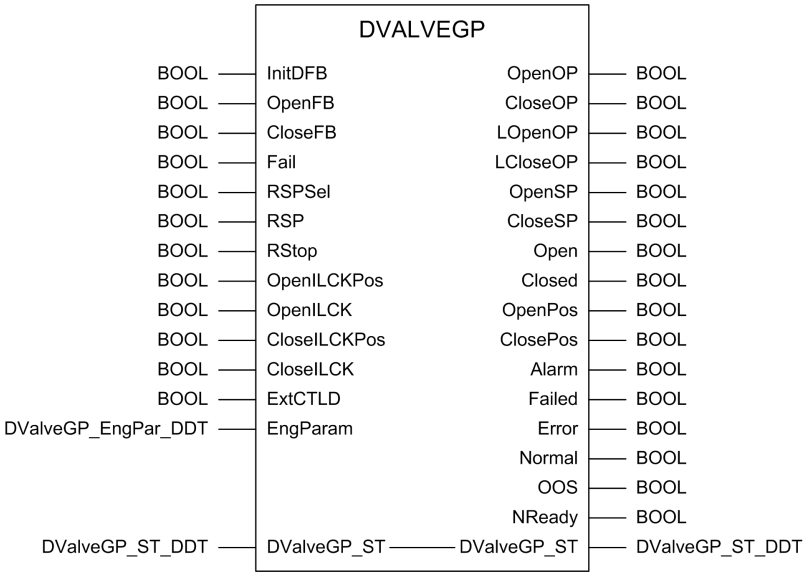
Overview

This section describes the `DVALVEGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
`InitDFB`	`BOOL`	If a rising edge is detected on this input, the internal timers of the block are reset and detected alarms are set to 0.
`OpenFB`	`BOOL`	1 = Indicates to the DFB that high limit position is reached. Usually, it is used to connect the digital input of a positioner high limit switch. The input pin is considered, if an input parameter is connected or linked.
`CloseFB`	`BOOL`	1 = Indicates to the DFB that low limit position is reached. Usually, it is used to connect the digital input of a positioner low limit switch. The input pin is considered, if an input parameter is connected or linked.
`Fail`	`BOOL`	1 = Indicates to the DFB a detected error in the device.
`RSPSel`	`BOOL`	Remote set point selection 1 = Low position set point 0 = High position set point
`RSP`	`BOOL`	Remote setpoint by continuous Control (0 = Close, 1 = Open). If `RStop` is connected, then the rising edge (Pulse based control) sends the command to start the motor only when program/remote cascade mode is configured.
`RStop`	`BOOL`	A rising edge sends a stop command only when program/remote cascade mode is configured.
`OpenILCKPos`	`BOOL`	Interlock position (1 = Open position will be considered when `OpenILCK` is active).
`OpenILCK`	`BOOL`	1 = Interlocks the device at the defined `OpenILCKPos` pin position (1 = Open interlock is active). **NOTE:** If the device requires resetting (`STW.ReamR` = 1), the `OP` takes the value = 0 regardless of the `OpenILCKPos` value waiting for the operator to carry out the reset.
`CloseILCK-Pos`	`BOOL`	Interlock position (1 = Close position will be considered when `CloseILCK` is active).

Parameter	Type	Description
CloseILCK	BOOL	1 = Interlocks the device at the defined CloseILCKPos pin position (1 = Close interlock is active). **NOTE:** If the device requires resetting (STW.ReamR = 1), the OP takes the value = 0 regardless of the CloseILCKPos value waiting for the operator to carry out the reset.
ExtCTLD	BOOL	1 = Indicates to the DFB whether the device is being controlled from outside of the system. Monitoring of confirmation alarms is stopped during external control. The detected error input and detected position error (OpenFB and CloseFB = 1) continues to be evaluated.
EngParam	DValveGP_EngPar_DDT	Engineering parameters

DValveGP_EngPar_DDT Type

Name	Type	Description
TimeOut	TIME	Enables to configure the maximum time that the valve takes to complete its stroke for closing and opening. When the output is triggered, if the limit switch is not detected even after the TimeOut time is elapsed. The device is switched to detected alarm state if the limit switch is not detected. When it has a value of 0, monitoring is deactivated.
ScanTime	TIME	The minimum time for which the detected alarm signals are kept active. ScanTime confirms that the supervision system acquires the data for detected alarms that are automatically reset.
ReactionTime	TIME	Enables to configure the time during which monitoring for the valve switching operations is enabled. During this time, a check is run to verify that the valve leaves the limit switch opposite to the operation that is carried out. When it has a value of 0, monitoring is disabled. This time is useful for detecting irregularity behavior in the valve and does not wait until the maximum stroke time is elapsed to detect them.
ConfRearmEn	BOOL	• 1 = Withdraws the output when a non-confirmed operation is detected (Alarm = 1) and requires a mandatory manual resetting once the Alarm condition disappears. • 0 = The output remains in its previous state when a non-confirmed operation is detected (Alarm = 1) and does not need manual resetting, once the Alarm condition disappears.
FailRearmEn	BOOL	• 1 = A manual resetting is required when the detected error/interlock trip alarm condition disappears. • 0 = No manual resetting is required when the detected error/interlock trip alarm condition disappears.
ResetSPFail	BOOL	Reset setpoint in case of detected error condition (1 = Restart is required if motor trips due to detected error and alarm).
TrackSPEn	BOOL	Track set-point in case of that the interlock condition is active. **NOTE:** The *TrackSPEn* parameter enables the interlock trip alarm when the device stopped due to interlock conditions.
Mode	BOOL	This input is used to select the operating modes of the DVALVEGP DFB. Initialization is triggered whenever there is a Mode change. Valid operating mode entry (0 = Continuous output, 3 = Position based, 4 = Position + Pulsetime.

Name	Type	Description
		NOTE: Mode change will not happen when DVALVEGP is in transition state, but it can happen in Open, Close, and Stop state.
SwitchDelay	TIME	This input is used to set a delay between output switching. Default value will be 0 seconds. For the delay duration set in this input, both the outputs will be low. Then after the delay, output (OpenOP or CloseOP) gets triggered based on the selected mode and input signal condition.
OpenOPPulse-Time	TIME	This input is used to set the pulse time duration for which the output OpenOP will be high, when there is a required setpoint. Once the pulse time duration elapses, the output OpenOP will go low and both the outputs remain low.
CloseOPPulse-Time	TIME	This input is used to set the pulse time duration for which output CloseOP will be high, when there is an required setpoint. Once the pulse time duration elapses, the output CloseOP will go low and both the outputs remain low.

Outputs

Output Parameter Description

Name	Type	Description
OpenOP	BOOL	1 = Field command signal for open position active. It always follows the LOpenOP command. **NOTE:** During simulation the signal will be deactivated.
CloseOP	BOOL	1 = Field command signal for close position active. It always follows the LCloseOP command. **NOTE:** During simulation the signal will be deactivated.
LOpenOP	BOOL	1 = Logical command signal for open position active. The current output (OpenOP output) is continuously copied to this variable except in the simulation mode. Refer to the LOpenOP and LCloseOP Calculation, page 303.
LClo-seOP	BOOL	1 = Logical command signal for close position active. The current output (CloseOP output) is continuously copied to this variable except in the simulation mode. Refer to the LOpenOP and LCloseOP Calculation, page 303.
OpenSP	BOOL	Open setpoint Refer to the SP Calculation, page 303.
CloseSP	BOOL	Close setpoint Refer to the SP Calculation, page 303.
Open	BOOL	1 = Indicates the maximum opening position is reached. It is calculated based on the states obtained from the analog input and/or the digital input belonging to the **open-valve limit** switch. It is evaluated by considering other detectors. Refer to the output Closed and input/output DValveGP_ST.CFGW.Sim, page 305. Refer to the Open and Closed Calculation, page 304.
Closed	BOOL	1 = Indicates the closed-valve position is reached. It is calculated based on the states obtained from the analog input and/or digital input belonging to the **closed-valve limit** switch. It is evaluated by considering other detectors. Refer to the output Open and input/output DValveGP_ST.CFGW.Sim, page 305. Refer to the Open and Closed Calculation, page 304.
OpenPos	BOOL	1 = Indicates the maximum opening position is reached. It is calculated based on the states obtained from the analog input and/or the digital input

Name	Type	Description
		belonging to the open valve limit switch. It is independent from the other detector if the CloseFB detector is available. Refer to the OpenPos and ClosePos calculation, page 305.
Close-Pos	BOOL	1 = Indicates the closed valve position is reached. It is calculated based on the states obtained from the analog input and/or the digital input belonging to the close valve limit switch. It is independent from the other detector if the OpenFB detector is available. Refer to the OpenPos and ClosePos calculation, page 305.
Alarm	BOOL	1 = The detection of unsuccessful operation has occurred.
Failed	BOOL	1 = Indicates an inoperable valve condition. It is evaluated according to the value of the Fail and FailEn inputs and detected position error (OpenFB and CloseFB) both are activated at same time. When it is activated, it is timed so that the signal is maintained for a minimum of ScanTime. 0 = After the minimum activation time has passed, it is only deactivated if the Fail input signal is deactivated or any one of the input limit switch is deactivated.
Error	BOOL	This output is generated when there is a detected configuration error in the DVALVEGP control function. • In MODE 4, Error detected is high if OpenOPPulseTime or CloseOPPulseTime is set as 0 seconds. • In MODE 4, Error detected is high if OpenFB and CloseFB input pins are not linked or assigned.
Normal	BOOL	Indicates the valve is operating in normal mode.
OOS	BOOL	1 = Indicates the valve is out of service.
NReady	BOOL	Indicates motor not ready for operation.

SP Calculation

The signal evaluation depending on the SC.Rem and RSP inputs and the DValveGP_ST.CFGW.Owner input/output is shown in the following table:

Owner	Rem	OpenSP is calculated as	CloseSP is calculated as
OFF	OFF	Sequential control (not SC.LSPSel and SC.LSP) or Pulse base control (SC.LOpen, SC.LStop)	Sequential control (SC.LSPSel and SC.LSP) or Pulse base control (SC.LClose, SC.LStop)
OFF	ON	Continuous control (not RSPSel, RSP) or Pulse base control if RStop pin is connected (not RSPSel, RSP, RStop)	Continuous control (not RSPSel, RSP) or Pulse base control if RStop pin is connected (RSPSel, RSP, RStop)
ON	-	Not DValveGP_ST.LSPSel and DValveGP_ST.LSP	DValveGP_ST.LSPSel and DValveGP_ST.LSP

LOpenOP and LCloseOP Calculation

The evaluation of the device control depending on the interlock status and if the DFB requires resetting is shown in the following table:

STW.RearmR	SC.ILCKD	LOpenOP is calculated as:	LCloseOP is calculated as:
ON	-	OFF	OFF
OFF	ON	Refer to the below Interlock calculation table.	
OFF	OFF	OpenSP	CloseSP

Interlock Calculation

OpenOP and CloseOP in interlock state (output will be ON until feedback is received or any detected error triggers):

When OpenOP = OFF and CloseOP = ON					
OpenILCK	OpenILCK-Pos	Close-ILCK	Close-ILCKPos	LOpenOP	LCloseOP
ON	OFF	OFF	OFF	Holds the last output state (Operator can change setpoint to Stop).	
ON	ON	OFF	OFF	ON	OFF
ON	OFF	ON	OFF	OFF	OFF
ON	OFF	ON	ON	OFF	ON
ON	ON	OFF	ON	ON	OFF
ON	ON	ON	ON	Holds the last output state.	
OFF	OFF	ON	OFF	OFF	OFF
OFF	OFF	ON	ON	OFF	ON
OFF	-	OFF	-	Holds the last output state.	

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

The *ConfRearmEn* engineering parameter has to be set to 1 for the *OP* pin of the function block output to become 0, under non-confirmed operations.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: Non-confirmed operations are those for which commands have been given and to which there is no confirming reply from the equipment.

Feedback Calculation for Open and Closed

OpenFB connected	CloseFB connected	Simulation	Override	Open is calculated as:	Closed is calculated as:
-	-	-	ON	LOpenOP	LCloseOP
-	-	ON	OFF	LOpenOP and (TimeOut/2)	LCloseOP and (TimeOut/2)
OFF	OFF	OFF	OFF	LOpenOP	LCloseOP
OFF	ON	OFF	OFF	Not CloseFB	CloseFB
ON	OFF	OFF	OFF	OpenFB	Not OpenFB
ON	ON	OFF	OFF	OpenFB and (not CloseFB)	CloseFB and (not OpenFB)

Feedback Calculation for OpenPos and ClosePos

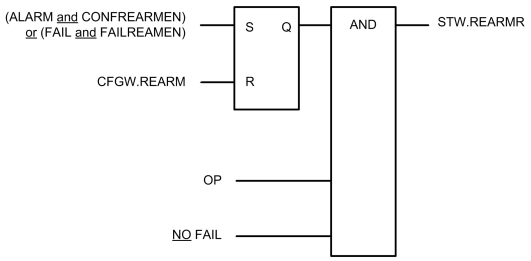
OpenFB connected	CloseFB connected	Simulation	Override	Open is calculated as:	Closed is calculated as:
-	-	-	ON	LOpenOP	LCloseOP
-	-	ON	OFF	LOpenOP and (TimeOut/2)	LCloseOP and (TimeOut/2)
OFF	OFF	OFF	OFF	LOpenOP	LCloseOP
OFF	ON	OFF	OFF	Not CloseFB	CloseFB
ON	OFF	OFF	OFF	OpenFB	Not OpenFB
ON	ON	OFF	OFF	OpenFB	CloseFB

Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
DValveGP_ST	DValveGP_ST_DDT	Data structure of the DFB status.

DValveGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	OpenPos	*Bit0*	BOOL	Refer to OpenPos output pin, page 302.
	ClosePos	*Bit1*	BOOL	Refer to ClosePos output pin, page 302.
	OpenSP	*Bit2*	BOOL	Refer to OpenSP output pin, page 302.
	CloseSP	*Bit3*	BOOL	Refer to CloseSP output pin, page 302.
	LOpenOP	*Bit4*	BOOL	Refer to LOpenOP output pin, page 302.
	LCloseOP	*Bit5*	BOOL	Refer to LCloseOP output pin, page 302.
	Rem	*Bit6*	BOOL	• 1 = Enables the DFB to be configured to remote set-point Rem. • 0 = Enables the DFB to be configured to local set-point LSP.
	ILCK	*Bit7*	BOOL	1= Interlock is active (OpenILCK or CloseILCK).
STW	RearmR	*Bit8*	BOOL	Write-only access Indicates if the valve requires resetting (1) after the detection of an inoperable device condition if it has been configured through the ConfRearmEn and/or FailRearmEn inputs. The signal activation logic included in the DFB is as follows. 
	ExtCTLD	*Bit9*	BOOL	Read-only access Refer to ExtCTLD input pin, page 300.
	OpenAlarm	*Bit10*	BOOL	1 = Open confirmation alarm active.

Name	Bit Name	Bit	Type	Description
	CloseAlarm	*Bit11*	BOOL	1 = Close confirmation alarm active.
	Failed	*Bit12*	BOOL	Refer to Failed output pin, page 302.
STW1	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	NReady	*Bit0*	BOOL	Refer to NReady output pin, page 302.
	Open-SPLocked	*Bit1*	BOOL	Indicates whether the open setpoint buttons in the monitoring (HMI) system has to be locked (1) or not (0).
	Close-SPLocked	*Bit2*	BOOL	Indicates whether the close setpoint buttons in the monitoring (HMI) system has to be locked (1) or not (0).
	OpenILCK-Trip	*Bit3*	BOOL	Interlock (1 = Open interlock trip active).
	CloseILCK-Trip	*Bit4*	BOOL	Interlock (1 = Close interlock trip active).
	OpenILCK	*Bit5*	BOOL	Interlock (1 = Open interlock is active).
	CloseILCK	*Bit6*	BOOL	Interlock (1 = Close interlock is active).
	FailAlarm	*Bit11*	BOOL	1 = FailAlarm is active.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read-write access to the data contained in this word.
	Owner	*Bit0*	BOOL	Read/write access Refer to Owner, page 306.
	ILCKBP	*Bit1*	BOOL	Read/write access 1 = Enables the interlock to be bypassed.
	Override	*Bit2*	BOOL	Refer to Override, page 306.
	Rearm	*Bit3*	BOOL	Write-only access Refer to Rearm, page 306.
	LSPSel	*Bit4*	BOOL	Refer to LSPSel, page 306.
	LSP	*Bit5*	BOOL	Read/write access Refer to LSP, page 306.
	SimFBFail	*Bit12*	BOOL	1 = Feedback will not simulate in simulation mode.
	Sim	*Bit13*	BOOL	Read/write access Refer to Sim, page 306.
	Maint	*Bit14*	BOOL	Refer to Maint, page 306.
	OOS	*Bit15*	BOOL	Read/write access Refer to OOS output pin, page 302.

Public Variables

Public Variable Description

Variable	Type	Description
SC	DValveGP_SC_DDT	Provides the frequently needed data to monitor the device status and to control it from the sequential control.

DValveGP_SC_DDT Type

Name	Type	Description
Rem	BOOL	Read/write access 1 = Enables the DFB to be configured to remote set-point RSP 0 = Enables the DFB to be configured to local set-point LSP.
LSPSel	BOOL	Read/write access Enables the sequential control to select the DFB to be affected by the local set-point. 1 = Corresponds to the low position direction. When the owner is the Program (Owner = 0) and the selected set-point is local (Rem = 0). 0 = Corresponds to the high position direction. When the owner is the Program (Owner = 0) and the selected set-point is local (Rem = 0). Calculation for LSPSel:

Owner	Rem	SC.LSPSel
OFF (Program)	OFF	Last value or new value assigned from the sequential control.
OFF	ON	Calculated from the RSPSel
ON (Operator)	ON	Last value or new value assigned by the Operator.

Name	Type	Description
LSP	BOOL	Read/write access 1 = Enables the assignment of the target local set-point usually for the sequential control if the owner is the Program (Owner input/output is OFF) and the selected set-point is Local (SC.Rem public variable is OFF). 0 = The block automatically assigns the current value of the resulting set-point. It is described as follows:

Owner	Rem	SC.LSP
OFF (Program)	OFF	SC.LSP (last value or new value assigned from the sequential control)
OFF	ON	SP (calculated from the RSP)
ON (Operator)	ON	SP (last value or new value assigned by the Operator)

Name	Type	Description
LOpen	BOOL	Read/write access 1 = Rising edge to enable Open command while the owner is in Program and Local.
LClose	BOOL	Read/write access 1 = Rising edge to enable Close command while the owner is in Program and Local.
LStop	BOOL	Read/write access 1 = Rising edge to enable Stop command while the owner is in Program and Local.
OpenOP	BOOL	Read-only access Refer to OpenOP output pin, page 302.
CloseOP	BOOL	Read-only access Refer to CloseOP output pin, page 302.
LOpenOP	BOOL	Read-only access Refer to LOpenOP output pin, page 302.
LCloseOP	BOOL	Read-only access

Name	Type	Description
		Refer to LCloseOP output pin, page 302.
OpenSP	BOOL	Read-only access Refer to OpenSP output pin, page 302.
CloseSP	BOOL	Read-only access Refer to CloseSP output pin, page 302.
Open	BOOL	Read-only access Refer to Open output pin, page 302.
Closed	BOOL	Read-only access Refer to Closed output pin, page 302.
OpenPos	BOOL	Read-only access Refer to OpenPos output pin, page 302.
ClosePos	BOOL	Read-only access Refer to ClosePos output pin, page 302.
OpenAlarm	BOOL	Read-only access 1 = Open position confirmation alarm active.
CloseAlarm	BOOL	Read-only access 1 = Close position confirmation alarm active.
Failed	BOOL	Read-only access Refer to Failed output pin, page 302.
FailAlarm	BOOL	Read-only access Refer to FailAlarm inputs/outputs pin, page 305.
OpenILCKD	BOOL	Read-only access 1 = Open interlocked and not bypassed.
CloseILCKD	BOOL	Read-only access 1 = Close interlocked and not bypassed.
ILCKD	BOOL	Read-only access The signal evaluation depending on the OpenILCKD and CloseILCKD input.
Rearm	BOOL	Read/write access Allows the device to be reset (1). The block sets the signal to 0 after each execution and sets the signal indicating that the resetting is required (CFGW.Rearm) to 0.
RearmR	BOOL	Read/write access Refer to RearmR, page 305 inputs/outputs pin.
Owner	REAL	Read/write access Enables to configure whether the set-point is set by the Program (0) or by the Operator (1).
Normal	BOOL	Read/write access Refer to Normal output pin, page 302.
Override	BOOL	Read/write access Enables to determine if the DFB is operating with external output (1 = Override) or not (0 = Normal).
Sim	BOOL	Read/write access Enables the simulation mode.
Maint	BOOL	Read/write access

Name	Type	Description
		Maintenance mode is active.
OOS	BOOL	Read/write access Refer to OOS output pin, page 302.
NReady	BOOL	Read/write access Refer to NReady output pin, page 302.
ExtCTLD	BOOL	Read/write access Refer to the ExtCTLD input pin, page 300.

Optional Service - Local Panel (DVALVELPGP)

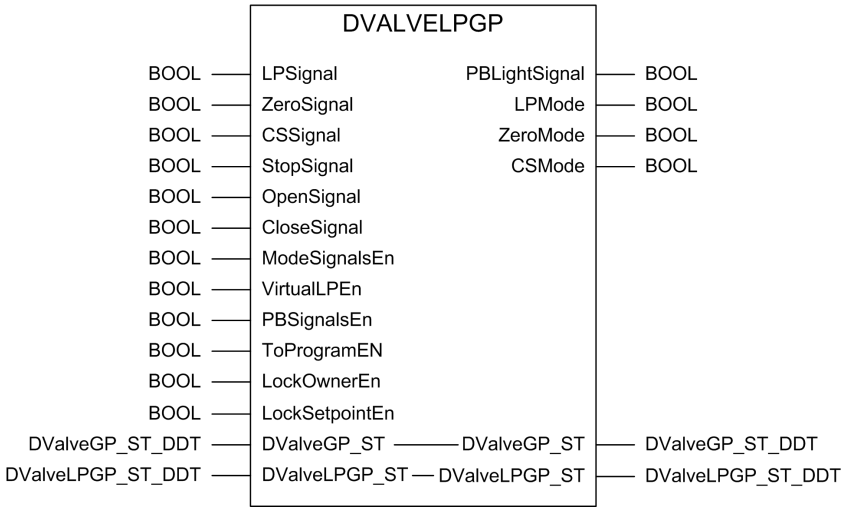
Overview

This chapter describes the optional services - local panel (DVALVELPGP) DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Name	Type	Description
LPSignal	BOOL	1 = The local panel mode switch is in the Local Panel position. Refer to the table included with the description of the LPMode output pin, page 311.
ZeroSignal	BOOL	1 = The local panel mode switch is in the zero position. This signal is optional even if a mode switch is available on the local panel. When this signal is a logical high, user is not allowed to operate the device from the faceplate and also from the DFB. Hence, all the functions in the faceplate will be disabled. Refer to the table included with the description of the LPMode output pin, page 311.

Name	Type	Description
CSSignal	BOOL	1 = The local panel mode switch is in the Control System position. Refer to the table included with the description of the LPMode output pin, page 311.
StopSignal	BOOL	1 = Stop signal from local panel.
OpenSignal	BOOL	1 = Indicates to the DFB the OPEN push button on the local panel is pressed.
CloseSignal	BOOL	1 = Indicates to the DFB the CLOSE push button on the local panel is pressed.
ModeSignalsEn	BOOL	1 = Enables the use of the Local Panel/Zero/ Control System (or Local Panel - Control System) mode switch on the local panel. Usually configured in development phase based on the characteristics of the local panel being used. If the mode switch is enabled, the LPSignal, ZeroSignal, and CSSignal inputs are considered to determine the operating modes of the local panel. Refer to the table included with the description of the LPMode output pin, page 311. **NOTE:** *ModeSignalsEn* input pin has higher priority than the *VirtualLPEn* input pin signal.
VirtualLPEn	BOOL	1 = This signal enables the operator to select the Local/Control System mode of operation from the HMI, so that the push button signals are enabled for operation. **NOTE:** When the owner is program, the Local/Control System mode selection drop-down list will be visible but disabled for operation however, when the owner is operator the Local Panel drop-down list will be accessible for operation. Refer to the table included with the description of the LPMode output pin, page 311.
PBSignalsEn	BOOL	1 = Enables to accept the OPEN and CLOSE push buttons from the local panel. This signal is not applicable if the mode switch is enabled (refer to the ModeSignalsEn input). This means that the push button signals are considered in the Local Panel mode. When the mode switch is disabled (ModeSignalsEn = 0), PBSignalsEn signal enables/disables the push button signals from control system itself based on relevant process conditions. SC.DisableLP public variable is also considered for determining whether the push buttons are enabled or not when *LPMode* pin signal is high as shown in the following table.

VirtualLPEn	ModeSignalsEn	PBSignal-sEn	SC.DisableLP	Push buttons enabled?
OFF	OFF	OFF	-	NO
OFF	OFF	-	ON	NO
OFF	OFF	ON	OFF	YES
OFF	ON	-	-	YES
ON	OFF	OFF	-	YES

Therefore, user can use this signal to enable/disable the push buttons on a simple local panel that does not feature a mode switch (Local Panel/Zero/Control System) based on the relevant process conditions.

Name	Type	Description
ToProgramEn	BOOL	1 = Change to program mode when control is switched to control system mode and VirtualLPEn = 0.
LockOwnerEn	BOOL	1 = Disables the access of owner of the valve remaining as Operator while the local panel mode is Local Panel. The following table shows how the valve owner is evaluated based on the ToProgramEn and LockOwnerEn inputs and operating mode of the local panel:

Local Panel Mode	ToProgra-mEn	LockOwne-rEn	Owner
Zero	–	–	Operator
Switch to Local Panel	–	–	Operator
Local Panel	–	ON	Operator
Switch to Control System	ON	–	Program

In remaining cases, owner of the valve ceases to be defined from the DVALVELPGP DFB, and it enables to change it from the monitoring (HMI) system with the command word in the DValveGP_ST.CFGW input/output of the corresponding DVALVEGP DFB.

In any case, consider that the owner of valve that had been set before switching to Local Panel mode is not memorized.

Name	Type	Description
LockSetpointEn	BOOL	1 = Disables the setpoint of the valve device and lock it into the one defined on the local panel while the local panel mode is Local Panel and the device owner is the Operator. You can switch to the Program mode based on the signals that have been previously described even if the local panel mode is set to Local Panel. Therefore, user can activate (1) this signal to lock setpoint operation from the monitoring (HMI) system while the valve is controlled from the local panel.

Outputs

Output Parameter Description

Name	Type	Description
PBLightSig-nal	BOOL	1 = The push buttons on the local panel are fully operational, that is, the local panel is in Local Panel mode (either because the mode is selected with the corresponding switch or because the push buttons are enabled in the event that there is no selector switch available or the *VirtualLPEn* is high with the local panel mode enabled).
LPMode	BOOL	1 = The local panel mode is in Local Panel. The following table shows how the local panel operating mode is determined based on the ModeSignalsEn, VirtualLPEn, PBSignalsEn, LPSignal, CSSignal and ZeroSignal, input signals:

Mode-Signal-sEn	Vir-tualLPEn	PBSigna-lsEn	LPSignal	CSSignal	ZeroSignal	OUTPUT
0	0	0	0	0	0	CS mode
1	0	0	0	0	0	Zero mode
1	0	0	1	0	0	LP mode and PBLightSignal
1	0	0	0	1	0	CS mode
0	1	1	NA	NA	NA	CS mode
0	1	0	NA	NA	NA	CS mode
0	0	1	0	0	0	LP mode and PBLightSignal
1	1	0	1	0	0	LP mode and PBLightSignal
1	1	0	0	0	0	Zero mode
1	1	0	0	0	1	Zero mode
1	1	0	0	1	0	CS mode
1	0	1	0	1	0	CS mode
1	1	1	0	1	0	CS mode
1	0	0	0	0	1	Zero mode

ZeroMode	BOOL	1 = The local panel mode is in Zero. User is not allowed to operate the device from the faceplate and also from DFB. Hence, all the functions in the faceplate will be disabled. Refer to the table included with the description of the LPMode output.
CSMode	BOOL	1 = The local panel mode is in Control System. Refer to the table included with the description of the LPMode output.

Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
DValveGP_ST	DValveGP_ST_DDT, page 305	Data structure belonging to the valve is to be controlled from the local panel (for detailed information regarding this structure, refer to the DVALVEGP DFB, page 300. The structure provides the information needed for the operation of the DVALVELPGP DFB and enables this DFB to control the owner of the valve.
DVal-veLPGP_ST	DValveLPGP_ST_DDT	Data structure that is used as an interface with the monitoring (HMI) system.

DValveLPGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	LPMode	0	BOOL	Refer to the LPMode output pin, page 311.
	ZeroMode	1	BOOL	Refer to the ZeroMode output pin, page 311.
	CSMode	2	BOOL	Refer to the CSMode output pin, page 311.
	OwnerLocked	3	BOOL	Indicates whether the owner change buttons in the monitoring (HMI) system needs to be locked (1) or not (0). Is activated when: • The local panel mode is set to Zero or • It is set to Local Panel, and the LockOwnerEn input signal has been configured as active (1).
	SPLocked	4	BOOL	Indicates whether the set-point change and START buttons in the monitoring (HMI) system needs to be locked (1) or not (0). Is activated when: • The local panel mode is set to Zero or • It is set to Local Panel, and the LockSetpointEn input signal has been configured as active (1).
	ModeSignalsEn	5	BOOL	Refer to the ModeSignalsEn input pin, page 309.
	PBEnabled	6	BOOL	Indicates whether the push buttons of the local panel are enabled (1) or not (0). The signal is activated when at least one of the ModeSignalsEn and PBSignalsEn signal inputs is active.
	VirtualLPEn	7	BOOL	Refer to the VirtualLPEn input pin, page 309.
	LPSignal	8	BOOL	Refer to the LPSignal input pin, page 309.
	ZeroSignal	9	BOOL	Refer to the ZeroSignal input pin, page 309.
	CSSignal	10	BOOL	Refer to the CSSignal input pin, page 309.
	StopSignal	11	BOOL	Refer to the StopSignal input pin, page 309.
	OpenSignal	12	BOOL	Refer to the OpenSignal input pin, page 309.
	CloseSignal	13	BOOL	Refer to the CloseSignal input pin, page 309.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
	LPEnabled	0	BOOL	Read/write access • 1 = Enables Local Panel mode. • 0 = Enables Control System mode.

Public Variables

Public Variable Description

Variable	Type	Description
SC	DValveLPGP_SC_DDT	Provides the frequently needed data for monitoring and controlling the DFB.

DValveLPGP_SC_DDT Type

Name	Type	Description
LPMode	BOOL	Read-only access Refer to the LPMode output pin, page 311.
ZeroMode	BOOL	Read-only access Refer to the ZeroMode output pin, page 311.
CSMode	BOOL	Read-only access Refer to the CSMode output pin, page 311.
DisableLP	BOOL	Read/write access 1 = Disables the push buttons on the local panel. Only applicable when a local panel without mode switch configuration is used (ModeSignalsEn = 0).

Supervision

Overview

This section describes the Supervision functions and runtime services that are available for the dual output valve.

Genies

Genie Properties

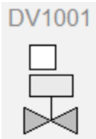
Refer to Genie Properties, page 62

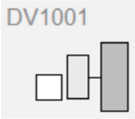
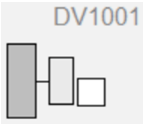
Genie Icons

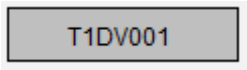
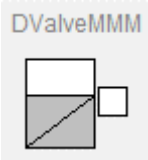
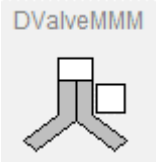
Refer to Genie Icons, page 63

Representation

The table describes the genies of the gpl_valve library:

Graphic symbol	Genie name	Description
	2way_Horiz	Horizontal two-way valve.
	2way_Vert	Vertical two-way valve.

Graphic symbol	Genie name	Description
	`Butterfly_Vert_Left`	Vertical butterfly with control on the left.
	`Butterfly_Vert_Right`	Vertical butterfly with control on the right.
	`3way_Horiz_DownLeft`	Horizontal three-way valve (down to left way when closed, down to right way when open).
	`3way_Horiz_DownRight`	Horizontal three-way valve (down to right way when closed, down to left way when open).
	`3way_Horiz_LeftDown`	Horizontal three-way valve (left to down way when closed, left to right way when open).
	`3way_Horiz_LeftRight`	Horizontal three-way valve (left to right way when closed, left to down way when open).
	`3way_Horiz_RightDown`	Horizontal three-way valve (right to down way when closed, right to left way when open).
	`3way_Horiz_RightLeft`	Horizontal three-way valve (right to left way when closed, right to down way when open).
	`3way_Vert_DownRight`	Vertical three-way valve (down to right way when closed, down to up way when open).
	`3way_Vert_DownUp`	Vertical three-way valve (down to up way when closed, down to right way when open).

Graphic symbol	Genie name	Description
	`3way_Vert_RightDown`	Vertical three-way valve (right to down way when closed, right to up way when open).
	`3way_Vert_RightUp`	Vertical three-way valve (right to up way when closed, right to down way when open).
	`3way_Vert_UpDown`	Vertical three-way valve (up to down way when closed, up to right way when open).
	`3way_Vert_UpRight`	Vertical three-way valve (up to right way when closed, up to down way when open).
	`valveinstanceid`	Dual actuator valve instance.
	`damper`	Valve - damper
	`diverter`	Valve - diverter

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

Available Tabs

During operation, clicking an dual output valve genie opens a faceplate with the following tabs:

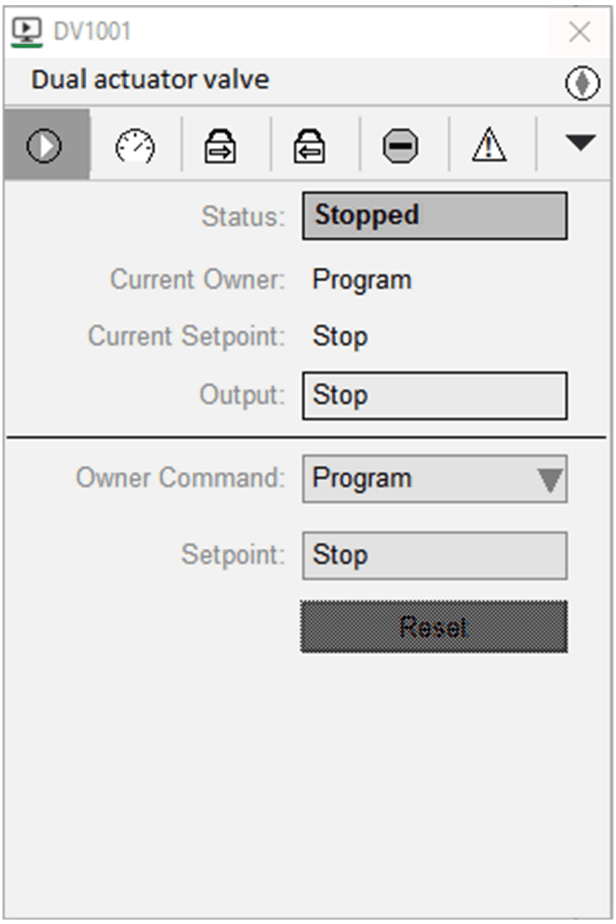
- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
  - Event, page 69

- Tabs for optional functions, which appear only if configured:
  - Interlock, page 87

    NOTE: Dual valve can have two interlock tabs, forward/reverse interlock.
  - Failure, page 78
  - Measures, page 70

Operator Tab

The figure shows an operator tab:



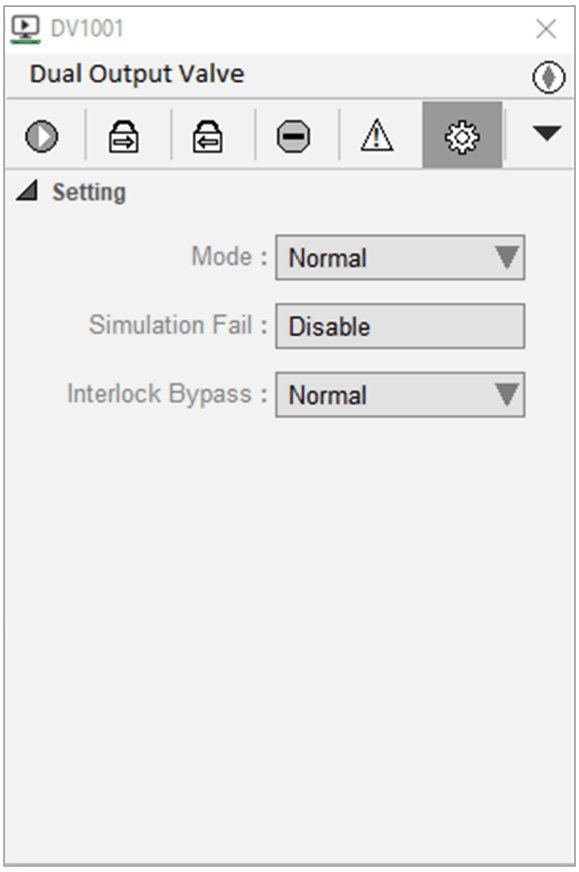
The table describes the control function of an operator tab:

Label	Item name	Description	Security parameter, page 297
Status	EquipStatus	Equipment status of dual output valve, For more details, refer to EquipStatus calculated variable, page 321.	–
Current Owner	Owner	Current owner of dual output valve, For more details, refer to operating owner, page 58.	
Current Setpoint	CurrentSetpoint	Current setpoint of dual output valve.	
Output	Equipstate	Output status of dual output valve, For more details, refer to EquipState calculated variable, page 321.	
OwnerCom-mand	OwnerCommand	Owner command of dual output valve (Program or Operator. **NOTE:** Local panel owner is enabled when the virtual panel is selected.	Set owner

Label	Item name	Description	Security parameter, page 297
Setpoint	CommandSet-point	Allows you to change the setpoint, if owner = operator.	Set setpoint
–	Rearm	Allows you to reset manually after the detection of an inoperable device condition or unsuccessful operation. **NOTE:** When the control module is reset, the current setpoint shown in operator tab is effective. Also, when you click **Reset**, a dialog box opens to confirm.	Set rearm

Engineering Tab

The figure shows an engineering tab:



The table describes the control function of engineering tab:

Label	Item name	Description	Security parameter, page 297
Mode	Mode	Allows you to view and change the mode, For more details, refer to operating mode, page 59.	Set mode - override, set mode - simulation, set mode - maintenance, and set mode - out of service **NOTE:** Each configurable mode has the different security parameters
Simulation Fail	SimulationFailActive	Enable or disable feedback simulation. **NOTE:** The Simulation Fail will only be available when the parameter `SimulationFail` is created in the citect parameters. To create, go to **Setup** > **Parameters**. Enter the `GPL.Mode` in **Section Name**, `SimulationFail` in **Name** and set the **Value** = 1.	Set mode - simulation
Interlock Bypass	InterlockBypass	Interlock bypass status Allows you to bypass interlock.	Set bypass conditions, set bypass all interlocks

Items

Overview

This section describes the variables, equipstatus and equipstate variable, abnormal and action required conditions, alarms, events, and trends items `dual actuator valve`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`FeedbackOpen`	Open feedback	Open/Not Open	*Bit0*
`FeedbackClosed`	Closed feedback	Closed/Not Closed	*Bit1*
`SetpointOpen`	Open setpoint selected	On/Off	*Bit2*
`SetpointClose`	Close setpoint selected		*Bit3*
`OutputOpen`	Open valve output status		*Bit4*
`OutputClose`	Close valve output status		*Bit5*
`ProgramSetpoint-Mode`	Setpoint source for program owner	Remote/Sequence	*Bit6*
`InterlockActive`	Interlock status	Interlocked/Healthy	*Bit7*
`RequireRearm`	Reset required	Required/Not Required	*Bit8*
`OwnerExternal`	External owner status	External/Not External	*Bit9*
`OpenAlarmActive`	Open feedback confirmation detected error	Active/Inactive	*Bit10*
`CloseAlarmActive`	Close feedback confirmation detected error		*Bit11*

Item name	Description	Enumeration	Address
FailActive	Valve inoperable		*Bit12*
Bit items derived from StatusWord1.			
NotReady	Not ready to function	–	*Bit0*
SetpointOpenLock-Active	Open setpoint change is denied		*Bit1*
SetpointCloseLock-Active	Close setpoint change is denied		*Bit2*
OpenInterlockTrip	Open interlock trip is active		*Bit3*
CloseInterlockTrip	Close interlock trip is active		*Bit4*
OpenInterlockAc-tive	Open interlocked is active		*Bit5*
CloseInterlockAc-tive	Close interlocked is active		*Bit6*
Bit items derived from ConfigurationWord.			
OwnerSelect	Owner of control module	Operator/Program	*Bit0*
InterlockBypass	Interlock bypass command	Bypassed/Normal	*Bit1*
InputSource	Input source selection	Operator/Field	*Bit2*
Rearm	Reset detected error	Active/Inactive	*Bit3*
DirectionSetpoint	Operator direction setpoint	Close/Open	*Bit4*
CommandSetpoint	Operator setpoint	On/Off	*Bit5*
SimulationFailAc-tive	Feedback detected error in simulation mode	–	*Bit12*
Simulation	Simulation mode		*Bit13*
Maintenance	Maintenance mode		*Bit14*
OutOfService	Valve unavailable	Out Of Service/In Service	*Bit15*
IO Items			
StatusWord	Status word	–	DVALVEGP_ST.STW
StatusWord1	Status word 1		DVALVEGP_ST.STW1
ConfigurationWord	Configuration word		DVALVEGP_ST.CFGW
Calculated Items			
EquipStatus	Current status of the dual output valve. For more details, refer to EquipStatus calculated variable, page 321.	–	GPL_ValveStatus
EquipState	Output information For more details, refer to EquipState calculated variable, page 321.		GPL_ ValveEquipState
Owner	Current owner of the dual output valve. For more details, refer to calculated variable, page 58.		GPL_ OwnerExternalLo-cal
Mode	Current mode of the dual output valve.		GPL_Mode

Item name	Description	Enumeration	Address
	For more details, refer to calculated variable, page 59.		
OperatorSetpoint	Operator setpoint of the dual output valve.		GPL_ DValveOperator-Setpoint
CurrentSetpoint	Current setpoint of the dual output valve.		GPL_ DValveCurrent-Setpoint
Abnormal	Shows abnormal condition of the dual output valve. For more details, refer to abnormal conditions, page 321.		GPL_Abnormal
ActionRequired	Shows action required condition of the dual output valve For more details, refer to action required conditions, page 322.		GPL_ ActionRequired
Optional Items			
Local panel			
Bit items derived from LPStatusWord.			
LocalPanelMode	Control from field	Enabled/Disabled	*Bit0*
ZeroMode	Motor control disabled		*Bit1*
ControlSystemMode	Control from remote		*Bit2*
LockedOwner	Owner change disabled from faceplate	Locked/Unlocked	*Bit3*
LockedSetpoint	Setpoint change disabled from faceplate		*Bit4*
LocalPanelEnabled	Control from field	Enabled/Disabled	*Bit5*
PushButtonEnabled	Local panel push buttons		*Bit6*
VirtualLocalPane-lEnabled	Local panel enable from control		*Bit7*
LocalPanelSelected	Control from field	Selected/Not Selected	*Bit8*
ZeroSelected	Control disabled		*Bit9*
ControlSystemSe-lected	Control from remote		*Bit10*
PushButtonStop	Stop push button signal	On/Off	*Bit11*
PushButtonOpen	Open push button signal		*Bit12*
PushButtonClose	Close push button signal		*Bit13*
Bit items derived from LPConfigurationWord.			
VirtualSwitch	Local panel selection from faceplate	Local/Control System	*Bit0*
IO Items			
LPStatusWord	Local panel status word	–	DVALVELPGP_ST. STW
LPConfiguration-Word	Local panel configuration word		DVALVELPGP_ST. CFGW

Item name	Description	Enumeration	Address
Failure	Refer to the topic describing the variable items of failures, page 61.		
Interlock	Refer to the topic describing the variable items of interlocks, page 60.		
Maintenance	Refer to the topic describing the variable items of maintenance, page 59.		

EquipStatus Calculated Variable

The table describes the *EquipStatus* calculated variable:

Equipment Status	Conditions	Enumeration
–	`Open` x `Closed`	Unknown
DValve Closed	Not `Open` x `Closed`	Closed
DValve Open	`Open` x Not `Closed`	Open
DValve Transition	`OP_OpenCmd` x Not `Open`	Opening
	`OP_CloseCmd` x Not `Closed`	Closing
DValve Stopped	Not `OP_OpenCmd` x Not `OP_CloseCmd` x Not `Open` x Not `Closed`	Stopped

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	Conditions	Enumeration
Passive with Out of Service	`OOS`	Stop
Active in Maintenance	Not `OOS` x `Maint` x `OpenCmd`	Open
Passive in Maintenance	Not `OOS` x `Maint` x `CloseCmd`	Close
Passive with Out of Service	Not `OOS` x `Maint` x Not `OpenCmd` x Not `CloseCmd`	Stop
Stop	Not `OOS` x Not `Maint` x `Fail`	Failed
Active with Interlock	Not `OOS` x Not `Maint` x Not `Fail` x `OpenCmd` x `OpenILCK`	Open Interlocked
Active	Not `OOS` x Not `Maint` x Not `Fail` x `OpenCmd` x Not `OpenILCK`	Open
Passive with Interlock	Not `OOS` x Not `Maint` x Not `Fail` x `CloseCmd` x `CloseILCK`	Close Interlocked
Active to Passive	Not `OOS` x Not `Maint` x Not `Fail` x `CloseCmd` x Not `CloseILCK`	Close
Passive with Interlock	Not `OOS` x Not `Maint` x Not `Fail` x Not `OpenCmd` x Not `CloseCmd` x (`OpenILCK` + `CloseILCK`)	Interlocked
Passive	Not `OOS` x Not `Maint` x Not `Fail` x Not `OpenCmd` x Not `CloseCmd` x Not `OpenILCK` x Not `CloseILCK`	Stop

Abnormal Conditions

The table describes the abnormal conditions of dual output valve:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
`AbnormalOwner`	Abnormal owner status appears, if the current owner is not a normal owner which configured at the instance level.	*Bit0*
`AbnormalMode`	Abnormal mode status appears, if the current operating mode is not normal mode.	*Bit1*

Item name	Description	Address
AbnormalInterlockBy-pass	Abnormal interlock bypass condition appears, if the Interlock bypass is enabled.	*Bit2*
AbnormalOperatorTab	Abnormal status of operator tab.	*Bit8*
AbnormalFailTab	Abnormal status of **Failures** tab.	*Bit10*
AbnormalEngineeringTab	Abnormal status of engineering tab.	*Bit12*
AbnormalDirection1In-terlockTab	Abnormal status of direction 1 interlock tab.	*Bit13*
AbnormalDirection2In-terlockTab	Abnormal status of direction 2 interlock tab.	*Bit14*

Action Required Conditions

The table describes the action required conditions of dual output valve:

Item name	Description	Address
Bit items derived from ActionRequired variable word.		
ActionOperatorTab	Action required status of operator tab.	*Bit8*
ActionFailTab	Action required status of **Failures** tab.	*Bit10*
ActionDirection1In-terlockTab	Action required status of action direction 1 interlock tab.	*Bit13*
ActionDirection1In-terlockTab	Action required status of action direction 2 interlock tab.	*Bit14*

Alarms

The table describes the advanced alarm items that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
VlvMntOperationHoursAlm	Running Hours Counter Overflow	GPL_BitValueCheck (StatusWord, 0)
VlvMntManeuversNumbersAlm	Operations Counter Overflow	GPL_BitValueCheck (StatusWord, 1)
Alarm	Detection of unsuccessful operation	GPL_BitValueCheck (StatusWord, 10,11,0)
Fail	Inoperable dual actuator valve	GPL_BitValueCheck (StatusWord, 12)
InterlockTrip	Interlock trip	GPL_BitValueCheck (StatusWord1, 3,4,0)

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
StateAlarm	All changes in the dual actuator valve status and output.
ModeAlarm	All changes in the mode of operation of the equipment.
ActionAlarm	All the operator actions in the supervision.

Trend Item

No trend item is used by Supervision components for `dual output valve`.

Motorized Valve

What's in This Chapter

Template 324
Control 329
Optional Service - Local Panel (MVALVEDLPGP) 338
Supervision 342

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the motorized valve.

Template

Overview

This section describes functionality of the `$MValveGP` template.

Description

General Description

The `$MValveGP` process control module template is designed to control a motorized valve or a gate, with two limit switches (open-valve and closed-valve) and two direction motor position switches.

Functional Description

The main functions of motorized valve template are described in the following table:

Function	Description
Control	Controls the digital control signal of the element based on the configuration and the commands that the template receives according to the functions.
Owner selection	The object manages the control system level, which is the owner (operator or program). As a result, it is responsible for setting the set-point for the desired position.
Interlocking	The template gives a command to the device to move to the defined position in case of an active interlock. An interlock bypass and setpoint is reset in case of interlock function is available.
Local and remote set-point	Enables to control the template with a set-point that is determined through a set-point selector (local or remote) in program owner. The local set-point is assigned from the sequential control while the remote set-point is assigned to the template from the logic implemented in the continuous control.
Mode selection	The object can be switched to different modes of operation like Simulation, Maintenance, Out Of Service and Normal.

Optional Services

Available optional services are:

- **Failures**

- **Interlocks**
- **Local Panel**
- **Maintenance**

Parameters

Parameters

The `$MValveGP` template defines operation of a typical motor valve. Alternative motor valve facets enable the modification of the motor valve that are available.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default value	Additional remarks
Setpoint	**Reset on failure**	`Boolean`	*FALSE*	Resets the setpoint when the motor is inoperable.
	Track setpoint on interlock	`Boolean`	*TRUE*	The interlock setpoint is tracked by the equipment setpoint when the interlock condition is active.
	Force remote setpoint	`Boolean`	*FALSE*	Enables the remote control operation for motor.
Failure	**Motor confirmation timeout**	`Duration`	*00:00:03*	Maximum time for feedback confirmation of the motor.
	Valve confirmation timeout		*00:00:10*	Maximum time for feedback confirmation of the valve.
	Maximum limit switch time		*00:00:02*	Maximum time to lose limit switch to confirm the start of operation.
	Minimum alarm duration		*00:00:05*	Minimum time to maintain the confirmation alarm or detected error alarm.
	Disable alarm when motor is not running	`Boolean`	*FALSE*	–
Rearm	**Required on failure**	`Boolean`	*TRUE*	Device requires a reset after device error or interlock trip alarm.
	Required on confirmation failure			Device requires a reset after confirmation error.
	Rearm resets conditions		*FALSE*	Rearms the motor, interlock and detected error conditions.

Inputs (Digital Input)

The table describes the digital input:

Name	Description
`OpenPosition`	It allows you to connect the open feedback of the valve.
`ClosePosition`	It allows you to connect the closed feedback of the valve.
`ForwardRunning`	It allows you to connect the forward direction feedback of the valve.
`ReverseRunning`	It allows you to connect the reverse direction feedback of the valve.

Name	Description
Fail	It allows you to connect the detected error condition of the motor.
ExternalControl	It allows you to control the device externally.

For more details, refer to the topic describing the parameters of digital inputs, page 44.

Outputs (Digital Output)

The table describes the digital output:

Name	Description
Open	It allows you to connect the digital output signal of valve to the external device.
Close	It allows you to disconnect the digital output signal of valve to the external device.

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Failures

For more details, refer to the topic describing the parameters of failures, page 74.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Local Panel (Logic)

For more details, refer to the topic describing the parameters of logic, page 45.

Maintenance

For more details, refer to the topic describing the parameters of maintenance, page 89.

Local Panel (Digital Input)

The table describes the digital input:

Name	Description
LPMode	Local panel selection signal from mode selector switch.
ZeroMode	Zero/Off selection signal from mode selector switch.
CSMode	Control selection signal from selector switch.
OffPB	Motor stop signal from local panel.
OpenPB	Motor start signal from local panel.
ClosePB	Motor stop signal from local panel.

For more details, refer to the topic describing the parameters of digital inputs, page 44.

Local Panel (Digital output)

The table describes the digital output:

Name	Description
PBLights	Output signal to indicate that push buttons are operational.

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default value	Additional remarks
Configu-ration	**Enable Separate Alarms for Open and Close Confirmation Failure**	Boo-lean	*FALSE*	It enables separate alarms for open and close confirmation error.
Opera-tion	**Normal Owner**	Enum	*Program and Cascade (5)*	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	**Set owner**	Enum	*Operator (1)*	Privilege to change owner. This parameter is used in Operator Tab, page 344.
	Set setpoint		*Operator (1)*	Privilege to change setpoint. This parameter is used in Operator Tab, page 344.
	Set mode - simulation		*Engineer (5)*	Privilege to change simulation mode. This parameter is used in Engineering Tab, page 345.
	Set mode - maintenance		*Maintenance (3)*	Privilege to change maintenance mode. This parameter is used in Engineering Tab, page 345.
	Set mode - out of service		*Supervisor (2)*	Privilege to change out of service mode. This parameter is used in Engineering Tab, page 345.
	Set reset		*Operator (Confirmed) (10)*	Privilege to reset. This parameter is used in Operator Tab, page 344.
	Set parameters		*Engineer (5)*	Privilege to change parameters. This parameter is used in Engineering Tab, page 345.
	Set alarm acknowledge		*Operator (1)*	Privilege to acknowledge alarm. This parameter is used in Alarm Tab, page 67.
	Set bypass conditions		*Supervisor (2)*	Privilege to set bypass conditions mode. This parameter is used in Engineering Tab, page 345.
	Set bypass all interlocks		*Supervisor (2)*	Privilege to bypass all interlocks. This parameter is used in Engineering Tab, page 345.
	Set asset maintenance		*Operator (1)*	Privilege to set maintenance mode. This parameter is used in Engineering Tab, page 345.
Events	**Enable log**	Boo-lean	*FALSE*	Enable event logging for asset status.
Historize	**Owner/mode**	Boo-lean	*FALSE*	It allows to historize the change in mode and owner.

Asset and Display Parameters

For more details, refer to the topic describing the parameters of asset and display, page 43.

Alarm Parameters

For more details, refer to the topic describing the parameters of alarm, page 45.

Failures

For more details, refer to the topic describing the parameters of failures, page 74.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Maintenance

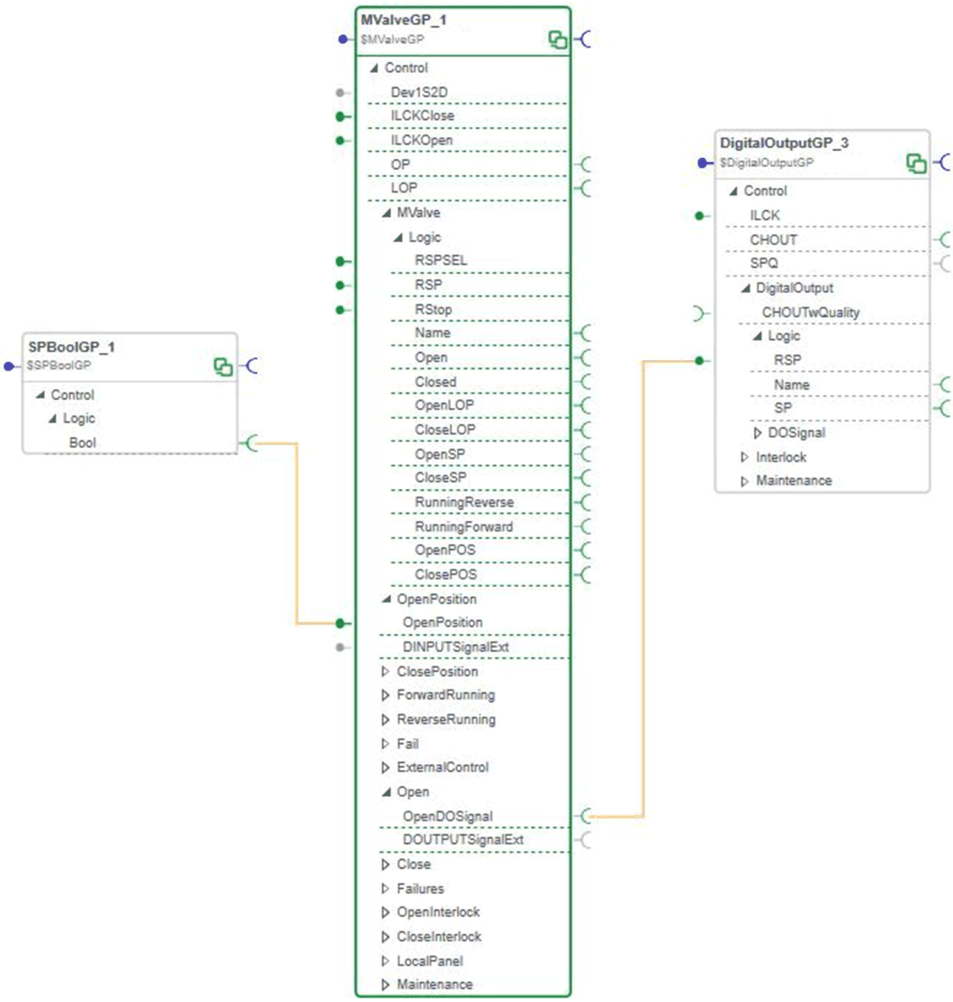
For more details, refer to the topic describing the parameters of maintenance, page 89.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

The following figure depicts the `$MValveGP` (also applicable for `$MValveCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



For interface identifiers, refer to the control module template interfaces, page 49.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

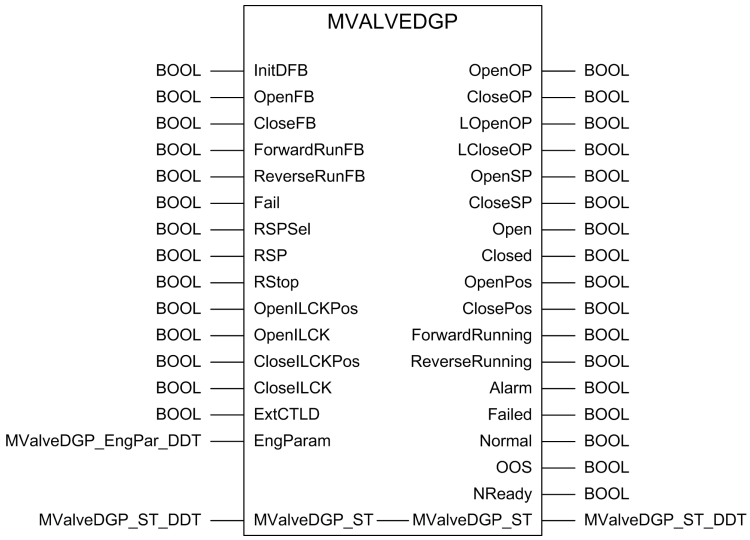
Overview

This section describes the `MVALVEDGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
`InitDFB`	`BOOL`	When a rising edge occurs on this input, the internal timers of the DFB are reset and alarms are set to 0.
`OpenFB`	`BOOL`	1 = Indicates that the open valve position is reached. Used to connect to the digital input of the open-valve limit switch. Disabled internally by the DFB if no variable or value is connected.
`CloseFB`	`BOOL`	1 = Indicates that the closed valve position is reached. Used to connect to the digital input of the closed-valve limit switch. Disabled internally by the DFB if no variable or value is connected.
`ForwardRunFB`	`BOOL`	1 = Indicates to the DFB that motor is running in forward direction. Usually, it is used to connect the digital input of a motor for switching operation confirmation. Disabled internally by the DFB if no variable or value is connected.
`ReverseRunFB`	`BOOL`	1 = Indicates to the DFB that motor is running in reverse direction. Usually, it is used to connect the digital input of a motor for switching operation confirmation. Disabled internally by the DFB if no variable or value is connected.
`Fail`	`BOOL`	1 = Indicates to the DFB a detected error in the device.
`RSPSel`	`BOOL`	Remote set point selection 1 = Low position set point 0 = High position set point

Parameter	Type	Description
RSP	BOOL	Remote setpoint by continuous Control (0 = Close, 1 = Open). If RStop is connected, then the rising edge (Pulse based control) sends the command to start the motor only when program/remote cascade mode is configured.
RStop	BOOL	A rising edge sends a stop command only when program/remote cascade mode is configured.
OpenILCKPos	BOOL	Interlock position (1 = Open position will be considered when OpenILCK is active).
OpenILCK	BOOL	1 = Interlocks the device at the defined OpenILCKPos pin position (1 = Open interlock is active). **NOTE:** If the device requires resetting (STW.ReamR = 1), the OP takes the value = 0 regardless of the OpenILCKPos value waiting for the operator to carry out the reset.
CloseILCKPos	BOOL	Interlock position (1 = Close position will be considered when CloseILCK is active).
CloseILCK	BOOL	1 = Interlocks the device at the defined CloseILCKPos pin position (1 = Close interlock is active). **NOTE:** If the device requires resetting (STW.ReamR = 1), the OP takes the value = 0 regardless of the CloseILCKPos value waiting for the operator to carry out the reset.
ExtCTLD	BOOL	1 = Indicates to the DFB whether the device is being controlled from outside of the system. Monitoring of confirmation alarms is stopped during external control. The detected error input and detected position error (OpenFB and CloseFB = 1) continues to be evaluated.

MValveDGP_EngPar_DDT Type

Name	Type	Description
TimeOutValve	TIME	Maximum time for position confirmation of valve (0s = Disabled).
TimeOutMotor	TIME	Maximum time for feedback confirmation of motor (0s = Disabled).
ScanTime	TIME	Enables the time that the detected alarm signals are kept active to be configured. Enables the monitoring subsystem to acquire the data for detected alarms that are automatically reset.
ReactionTime	TIME	Enables to configure the time during which monitoring for the valve switching operations is enabled. During this time, a check is run to verify that the valve leaves the limit switch opposite to the operation that is carried out. When it has a value of 0, monitoring is disabled. This time is useful for detecting irregularity behavior in the valve and does not wait until the maximum stroke time is elapsed to detect them.
ConfRearmEn	BOOL	• 1 = Withdraws the output when a non-confirmed operation is detected (Alarm = 1) and requires a mandatory manual resetting once the Alarm condition disappears. • 0 = The output remains in its previous state when a non-confirmed operation is detected (Alarm = 1) and does not need manual resetting, once the Alarm condition disappears.
FailRearmEn	BOOL	• 1 = A manual resetting is required when the detected error/interlock trip alarm condition disappears. • 0 = No manual resetting is required when the detected error/interlock trip alarm condition disappears.
DisStopAlarm	BOOL	Disable device detected error alarm when motor is not running.
ResetSPFail	BOOL	Reset setpoint in case of detected error condition (1 = Restart is required if motor trips due to detected error and alarm).
TrackSPEn	BOOL	Track set-point in case of that the interlock condition is active. **NOTE:** The *TrackSPEn* parameter enables the interlock trip alarm when the device stopped due to interlock conditions.

Outputs

Output Parameter Description

Name	Type	Description
OpenOP	BOOL	1 = Field command signal for open position active. It always follows the LOpenOP command. **NOTE:** During simulation the signal will be deactivated.
CloseOP	BOOL	1 = Field command signal for close position active. It always follows the LCloseOP command. **NOTE:** During simulation the signal will be deactivated.
LOpenOP	BOOL	1 = Logical command signal for open position active. The current output (OpenOP output) is continuously copied to this variable except in the simulation mode. Refer to the LOpenOP and LCloseOP Calculation, page 332.
LCloseOP	BOOL	1 = Logical command signal for close position active. The current output (CloseOP output) is continuously copied to this variable except in the simulation mode. Refer to the LOpenOP and LCloseOP Calculation, page 332.
OpenSP	BOOL	Open setpoint Refer to the SP Calculation, page 332.
CloseSP	BOOL	Close setpoint Refer to the SP Calculation, page 332.
Open	BOOL	1 = Indicates the maximum opening position is reached. It is calculated based on the states obtained from the analog input and/or the digital input belonging to the **open-valve limit** switch. It is evaluated by considering other detectors. Refer to the output Closed and input/output MValveDGP_ST.CFGW.Sim, page 334. Refer to the Open and Closed calculation, page 333.
Closed	BOOL	1 = Indicates the closed-valve position is reached. It is calculated based on the states obtained from the analog input and/or digital input belonging to the **closed-valve limit** switch. It is evaluated by considering other detectors. Refer to the output Open and input/output MValveDGP_ST.CFGW.Sim, page 334. Refer to the Open and Closed calculation, page 333.
OpenPos	BOOL	1 = Indicates the maximum opening position is reached. It is calculated based on the states obtained from the analog input and/or the digital input belonging to the open valve limit switch. It is independent from the other detector if the CloseFB detector is available. Refer to the OpenPos and ClosePos calculation, page 333.
ClosePos	BOOL	1 = Indicates the closed valve position is reached. It is calculated based on the states obtained from the analog input and/or the digital input belonging to the close valve limit switch. It is independent from the other detector if the OpenFB detector is available. Refer to the OpenPos and ClosePos calculation, page 333.
ForwardRunning	BOOL	1 = The motor has reached the position corresponding to setpoint. It is evaluated by considering any detectors. Refer to the ReverseRunning output and MValveDGP_ST.CFGW.Sim, page 334 input/output pin. Refer to the ForwardRunning and ReverseRunning calculation, page 333.
ReverseRunning	BOOL	1 = The motor has reached the position corresponding to setpoint. It is evaluated by considering any detectors. Refer to the ForwardRunning output and MValveDGP_ST.CFGW.Sim, page 334 input/output pin. Refer to the ForwardRunning and ReverseRunning calculation, page 333.
Alarm	BOOL	1 = The detection of unsuccessful operation has occurred.
Failed	BOOL	1 = Irregularity in one of the auxiliary devices.
Normal	BOOL	Indicates the valve is operating in normal mode.

Name	Type	Description
OOS	BOOL	1 = Indicates the valve is out of service.
NReady	BOOL	Indicates motor not ready for operation.

SP Calculation

The signal evaluation depending on the SC.Rem and RSP inputs and the MValveDGP_ST.CFGW.Owner input/output is shown in the following table:

Owner	Rem	OpenSP is calculated as	CloseSP is calculated as
OFF	OFF	Sequential control (not SC.LSPSel) and SC.LSP) or Pulse base control (SC.LOpen, SC.LStop)	Sequential control (SC.LSPSel) and SC.LSP) or Pulse base control (SC.LClose, SC.LStop)
OFF	ON	Continuous control (not RSPSel), RSP) or Pulse base control if RStop pin is connected (not RSPSel, RStop)	Continuous control (not RSPSel), RSP) or Pulse base control if RStop pin is connected (RSPSel, RSP, RStop)
ON	-	Not MValveDGP_ST.LSPSel and MValveDGP_ST.LSP	MValveDGP_ST.LSPSel and MValveDGP_ST.LSP

LOpenOP and LCloseOP Calculation

The evaluation of the device control depending on the interlock status and if the DFB requires resetting is shown in the following table:

STW.RearmR	SC.ILCKD	LOpenOP is calculated as:	LCloseOP is calculated as:
ON	-	OFF	OFF
OFF	ON	Refer to the below Interlock calculation table.	
OFF	OFF	OpenSP	CloseSP

Interlock Calculation

OpenOP and CloseOP in interlock state (output will be ON until feedback is received or any detected error triggers):

When OpenOP = OFF and CloseOP = ON					
OpenILCK	Open-ILCKPos	Close-ILCK	Close-ILCKPos	LOpenOP	LCloseOP
ON	OFF	OFF	OFF	Holds the last output state (Operator can change setpoint to Stop).	
ON	ON	OFF	OFF	ON	OFF
ON	OFF	ON	OFF	OFF	OFF
ON	OFF	ON	ON	OFF	ON
ON	ON	OFF	ON	ON	OFF
ON	ON	ON	ON	Holds the last output state.	
OFF	OFF	ON	OFF	OFF	OFF
OFF	OFF	ON	ON	OFF	ON
OFF	-	OFF	-	Holds the last output state.	

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

The *ConfRearmEn* engineering parameter has to be set to 1 for the *OP* pin of the function block output to become 0, under non-confirmed operations.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: Non-confirmed operations are those for which commands have been given and to which there is no confirming reply from the equipment.

Feedback Calculation for `Open`, `Closed`, `OpenPos`, `ClosePos`,

`Simulation`	`Open`	`Closed`	`OpenPos` is calculated as:	`ClosePos` is calculated as:
ON	`LOpenOP` and (`TimeOut`/2)	`LCloseOP` and (`TimeOut`/2)	`OpenOP` and (`TimeOut`/2)	`CloseOP` and (`TimeOut`/2)
OFF	`OpenFB` and (not `CloseFB`)	`CloseFB` and (not `OpenFB`)	`OpenFB`	`CloseFB`

Feedback Calculation for `ForwardRunning` and `ReverseRunning`

`ExtCTLD`	`ForwardRunningFB` connected	`ReverseRunningFB` connected	`Simulation`	`ForwardRunning` is calculated as:	`ReverseRunning` is calculated as:
-	-	-	ON	`LOpenOP` and (`TimeOut`/2)	`LCloseOP` and (`TimeOut`/2)
OFF	OFF	OFF	OFF	`LOpenOP`	`LCloseOP`
ON	OFF	OFF	OFF	OFF	OFF
OFF	OFF	ON	OFF	`ReverseRunningFB` and `OpenOP`	`ReverseRunningFB` and `CloseOP`
OFF	ON	OFF	OFF	`ForwardRunningFB` and `OpenOP`	`ForwardRunningFB` and `CloseOP`
OFF	ON	ON	OFF	`ForwardRunningFB` and (not `ReverseRunningFB`	`ReverseRunningFB` and (not `ForwardRunningFB`
ON	ON	OFF	OFF	OFF	OFF
ON	OFF	ON	OFF	OFF	OFF

Feedback Calculation for `ForwardRunningFB` and `ReverseRunningFB`

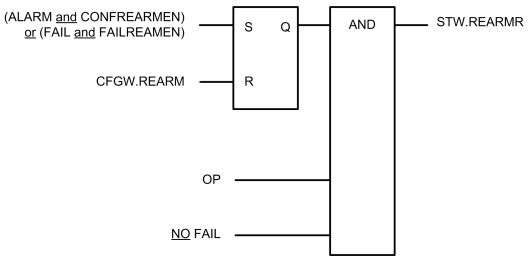
`ExtCTLD`	`ForwardRunningFB` connected	`ReverseRunningFB` connected	`Simulation`	`ForwardRunningFB` in `ST` is calculated as:	`ReverseRunningFB` in `ST` is calculated as:
-	-	-	ON	`LOpenOP` and (`TimeOut`/2)	`LCloseOP` and (`TimeOut`/2)
OFF	OFF	OFF	OFF	`LOpenOP`	`LCloseOP`
ON	OFF	OFF	OFF	OFF	OFF
OFF	OFF	ON	OFF	`ReverseRunningFB` and `OpenOP`	`ReverseRunningFB` and `CloseOP`
OFF	ON	OFF	OFF	`ForwardRunningFB` and `OpenOP`	`ForwardRunningFB` and `CloseOP`
OFF	ON	ON	OFF	`ForwardRunningFB`	`ReverseRunningFB`
ON	ON	OFF	OFF	OFF	OFF
ON	OFF	ON	OFF	OFF	OFF

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
MValveDGP_ST	MValveDGP_ST_DDT	Provides the data needed to monitor and/or control the block status.

MValveDGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	OpenPos	*Bit0*	BOOL	Refer to OpenPos output pin, page 331.
	ClosePos	*Bit1*	BOOL	Refer to ClosePos output pin, page 331.
	OpenSP	*Bit2*	BOOL	Refer to OpenSP output pin, page 331.
	CloseSP	*Bit3*	BOOL	Refer to CloseSP output pin, page 331.
	LOpenOP	*Bit4*	BOOL	Refer to LOpenOP output pin, page 331.
	LCloseOP	*Bit5*	BOOL	Refer to LCloseOP output pin, page 331.
	Rem	*Bit6*	BOOL	• 1 = Enables the DFB to be configured to remote set-point Rem. • 0 = Enables the DFB to be configured to local set-point LSP.
	ILCK	*Bit7*	BOOL	1= Interlock is active (OpenILCK or CloseILCK).
	RearmR	*Bit8*	BOOL	Indicates if the valve requires resetting (1) after the detection of an inoperable device condition if it has been configured through the ConfRearmEn and/or FailRearmEn inputs. The signal activation logic included in the DFB is as follows. 
STW	ExtCTLD	*Bit9*	BOOL	Refer to ExtCTLD input pin, page 329.
	OpenAlarm	*Bit10*	BOOL	Refer to OpenAlarm, page 335.
	CloseAlarm	*Bit11*	BOOL	Refer to CloseAlarm, page 335.
	Failed	*Bit12*	BOOL	Refer to Failed output pin, page 331.
	ForwardA-larm	*Bit13*	BOOL	Refer to ForwardAlarm, page 335.
	ReverseA-larm	*Bit14*	BOOL	Refer to ReverseAlarm, page 335.
STW1	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	NReady	*Bit0*	BOOL	Refer to NReady output pin, page 331.
	Open-SPLocked	*Bit1*	BOOL	Indicates whether the open setpoint buttons in the monitoring (HMI) system has to be locked (1) or not (0).
	Close-SPLocked	*Bit2*	BOOL	Indicates whether the close setpoint buttons in the monitoring (HMI) system has to be locked (1) or not (0).

Name	Bit Name	Bit	Type	Description
	OpenILCK-Trip	*Bit3*	BOOL	1 = Open interlock trip active.
	CloseILCK-Trip	*Bit4*	BOOL	1 = Close interlock trip active.
	OpenILCK	*Bit5*	BOOL	Refer to OpenILCK input pin, page 329.
	CloseILCK	*Bit6*	BOOL	Refer to CloseILCK input pin, page 329.
	ForwardRun-ningFB	*Bit8*	BOOL	Refer to ForwardRunningFB input pin, page 329.
	ReverseRun-ningFB	*Bit9*	BOOL	Refer to ReverseRunningFB input pin, page 329.
	FailAlarm	*Bit11*	BOOL	1 = FailAlarm is active.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read-write access to the data contained in this word.
	Owner	*Bit0*	BOOL	Refer to Owner, page 335.
	ILCKBP	*Bit1*	BOOL	1 = Enables the interlock to be bypassed.
	Rearm	*Bit3*	BOOL	Write-only access Refer to Rearm, page 335.
	LSPSel	*Bit4*	BOOL	Read/write access Refer to LSPSel, page 335.
	LSP	*Bit5*	BOOL	Read/write access Refer to LSP, page 335.
	SimFBFail	*Bit12*	BOOL	1 = Feedback will not simulate in simulation mode.
	Sim	*Bit13*	BOOL	Refer to Sim, page 335.
	Maint	*Bit14*	BOOL	Refer to Maint, page 335.
	OOS	*Bit15*	BOOL	Refer to OOS output pin, page 331.

Public Variables

Public Variable Description

Variable	Type	Description
SC	MValveDGP_SC_DDT	Provides the common data required to monitor and control the valve status from the sequential control.

MValveDGP_SC_DDT Type

Name	Type	Description
Rem	BOOL	Read/write access 1 = Enables the DFB to be configured to remote set-point RSP 0 = Enables the DFB to be configured to local set-point LSP.
LSPSel	BOOL	Read/write access Enables the sequential control to select the actuator to be affected by the local set-point. 1 = Corresponds to the closing direction.when the owner is the Program (OWNER = 0) and the selected set-point is local (REM = 0). 1 = Corresponds to the opening direction.when the owner is the Program (OWNER = 0) and the selected set-point is local (REM = 0).

Name	Type	Description
		Calculation for `LSPSel`:
		`Owner` / **`Rem`** / **`SC.LSP`**
		OFF (Program) / OFF / `SC.LSP` (last value or new value assigned from the sequential control)
		OFF / ON / `SP` (calculated from the `RSP` input)
		ON (Operator) / ON / `SP` (last value or new value assigned by the Operator)
`LSP`	`BOOL`	Read/write access Enables the sequential control the assignment of the local set-point if the owner is the Program (OWNER = 0) and the selected set-point is local (`REM` public variable = 0). 0 = The block automatically assigns the current value of the resulting set-point. It is described as follows:
		`Owner` / **`Rem`** / **`SC.LSP`**
		OFF (Program) / OFF / `SC.LSP` (last value or new value assigned from the sequential control)
		OFF / ON / `SP` (calculated from the `RSP` input)
		ON (Operator) / ON / `SP` (last value or new value assigned by the Operator)
`LOpen`	`BOOL`	Read/write access 1 = Rising edge to enable `Open` command while the owner is in `Program` and `Local`.
`LClose`	`BOOL`	Read/write access 1 = Rising edge to enable `Close` command while the owner is in `Program` and `Local`.
`LStop`	`BOOL`	Read/write access 1 = Rising edge to enable `Stop` command while the owner is in `Program` and `Local`.
`OpenOP`	`BOOL`	Read-only access Refer to `OpenOP` output pin, page 331.
`CloseOP`	`BOOL`	Read-only access Refer to `CloseOP` output pin, page 331.
`LOpenOP`	`BOOL`	Read-only access Refer to `LOpenOP` output pin, page 331.
`LCloseOP`	`BOOL`	Read-only access Refer to `LCloseOP` output pin, page 331.
`OpenSP`	`BOOL`	Read-only access Refer to `OpenSP` output pin, page 331.
`CloseSP`	`BOOL`	Read-only access Refer to `CloseSP` output pin, page 331.
`Open`	`BOOL`	Read-only access Refer to `Open` output pin, page 331.
`Closed`	`BOOL`	Read-only access Refer to `Closed` output pin, page 331.

Name	Type	Description
OpenPos	BOOL	Read-only access Refer to OpenPos output pin, page 331.
ClosePos	BOOL	Read-only access Refer to ClosePos output pin, page 331.
ForwardRun-ningFB	BOOL	Read-only access Refer to ForwardRunningFB input pin, page 329.
ReverseRun-ningFB	BOOL	Read-only access Refer to ReverseRunningFB input pin, page 329.
OpenAlarm	BOOL	Read-only access 1 = Open position confirmation alarm active (Read-only).
CloseAlarm	BOOL	Read-only access 1 = Close position confirmation alarm active (Read-only).
ForwardAlarm	BOOL	Read-only access 1 = Motor forward confirmation alarm active (Read-only).
ReverseAlarm	BOOL	Read-only access 1 = Motor reverse confirmation alarm active (Read-only).
Failed	BOOL	Read-only access Refer to Failed output pin, page 331.
FailAlarm	BOOL	Read-only access Refer to FailAlarm inputs/outputs pin, page 334.
OpenILCKD	BOOL	Read-only access 1 = Open interlocked and not bypassed (Read-only).
CloseILCKD	BOOL	Read-only access 1 = Close interlocked and not bypassed (Read-only).
ILCKD	BOOL	Read-only access The signal evaluation depending on the OpenILCKD and CloseILCKD input.
Rearm	BOOL	Read/write access Allows the device to be reset (1). The block sets the signal to 0 after each execution and sets the signal indicating that the resetting is required (CFGW.Rearm) to 0.
RearmR	BOOL	Read/write access Refer to RearmR inputs/outputs pin, page 334.
Owner	REAL	Read/write access Enables to configure whether the set-point is set by the Program (0) or by the Operator (1).
Normal	BOOL	Read-only access Refer to Normal output pin, page 331.
Sim	BOOL	Read/write access Enables the simulation mode.
Maint	BOOL	Read-only access Maintenance mode is active.
OOS	BOOL	Read/write access Refer to OOS output pin, page 331.

Name	Type	Description
NReady	BOOL	Read-only access Refer to NReady output pin, page 331.
ExtCTLD	BOOL	Read-only access Refer to the ExtCTLD input pin, page 329.

Optional Service - Local Panel (MVALVEDLPGP)

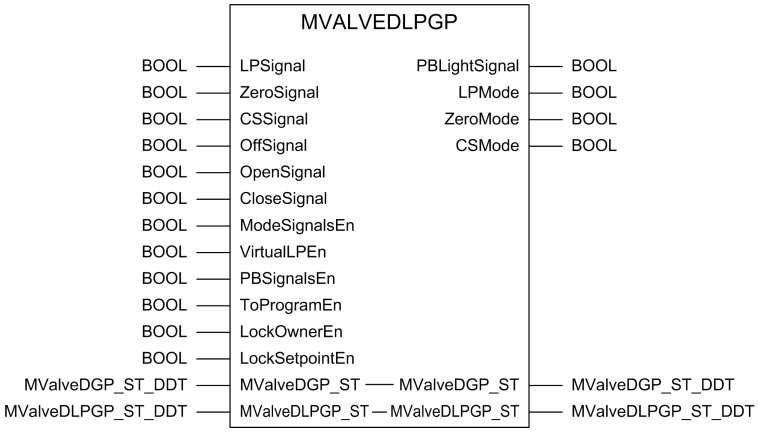
Overview

This chapter describes the optional services - local panel (MVALVEDLPGP) DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Name	Type	Description
LPSignal	BOOL	1 = The local panel mode switch is in the Local Panel position. Refer to the table included with the description of the LPMode output pin, page 340.
ZeroSignal	BOOL	1 = The local panel mode switch is in the zero position.This signal is optional even if a mode switch is available on the local panel. When this signal is a logical high, user is not allowed to operate the device from the faceplate and also from the DFB. Hence, all the functions in the faceplate will be disabled. Refer to the table included with the description of the LPMode output pin, page 340.
CSSignal	BOOL	1 = The local panel mode switch is in the Control System position. Refer to the table included with the description of the LPMode output pin, page 340.
OffSignal	BOOL	1 = The OFF push button on the local panel is pressed.
OpenSignal	BOOL	1 = Indicates to the DFB the OPEN push button on the local panel is pressed.
CloseSignal	BOOL	1 = Indicates to the DFB the CLOSE push button on the local panel is pressed.
ModeSignalsEn	BOOL	1 = Enables the use of the Local Panel/Zero/ Control System (or Local Panel - Control System) mode switch on the local panel. Usually configured in development phase based on the characteristics of

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td></td><td></td><td>the local panel being used. If the mode switch is enabled, the LPSignal, ZeroSignal, and CSSignal inputs are considered to determine the operating modes of the local panel. Refer to the table included with the description of the LPMode output pin, page 340.

NOTE: ModeSignalsEn input pin has higher priority than the VirtualLPEn input pin signal.</td></tr>
<tr><td>VirtualLPEn</td><td>BOOL</td><td>1 = This signal enables the operator to select the Local/Control System mode of operation from the HMI, so that the push button signals are enabled for operation.

NOTE: When the owner is program, the Local/Control System mode selection drop-down list will be visible but disabled for operation however, when the owner is operator the Local Panel drop-down list will be accessible for operation.

Refer to the table included with the description of the LPMode output pin, page 340.</td></tr>
<tr><td>PBSignalsEn</td><td>BOOL</td><td>1 = Enables to accept the OPEN and CLOSE push buttons from the local panel. This signal is not applicable if the mode switch is enabled (refer to the ModeSignalsEn input). This means that the push button signals are considered in the Local Panel mode.

When the mode switch is disabled (ModeSignalsEn = 0), PBSignalsEn signal enables/disables the push button signals from control system itself based on relevant process conditions. SC.DisableLP public variable is also considered for determining whether the push buttons are enabled or not when LPMode pin signal is high as shown in the following table.
<table>
<tr><th>VirtualLPEn</th><th>ModeSignalsEn</th><th>PBSignal-sEn</th><th>SC.DisableLP</th><th>Push buttons enabled?</th></tr>
<tr><td>OFF</td><td>OFF</td><td>OFF</td><td>-</td><td>NO</td></tr>
<tr><td>OFF</td><td>OFF</td><td>-</td><td>ON</td><td>NO</td></tr>
<tr><td>OFF</td><td>OFF</td><td>ON</td><td>OFF</td><td>YES</td></tr>
<tr><td>OFF</td><td>ON</td><td>-</td><td>-</td><td>YES</td></tr>
<tr><td>ON</td><td>OFF</td><td>OFF</td><td>-</td><td>YES</td></tr>
</table>
Therefore, user can use this signal to enable/disable the push buttons on a simple local panel that does not feature a mode switch (Local Panel/Zero/Control System) based on the relevant process conditions.</td></tr>
<tr><td>ToProgramEn</td><td>BOOL</td><td>1 = Change to program mode when control is switched to control system mode and VirtualLPEn = 0.</td></tr>
<tr><td>LockOwnerEn</td><td>BOOL</td><td>1 = Disables the access of owner of the valve remaining as Operator while the local panel mode is Local Panel.

The following table shows how the valve owner is evaluated based on the ToProgramEn and LockOwnerEn inputs and operating mode of the local panel:
<table>
<tr><th>Local Panel Mode</th><th>ToProgra-mEn</th><th>LockOwne-rEn</th><th>Owner</th></tr>
<tr><td>Zero</td><td>–</td><td>–</td><td>Operator</td></tr>
<tr><td>Switch to Local Panel</td><td>–</td><td>–</td><td>Operator</td></tr>
<tr><td>Local Panel</td><td>–</td><td>ON</td><td>Operator</td></tr>
<tr><td>Switch to Control System</td><td>ON</td><td>–</td><td>Program</td></tr>
</table>
In remaining cases, owner of the valve ceases to be defined from the MVALVEDLPGP DFB, and it enables to change it from the monitoring (HMI) system with the command word in the MValveDGP_ST.CFGW input/output of the corresponding MVALVEDGP DFB.

In any case, consider that the owner of valve that had been set before switching to Local Panel mode is not memorized.</td></tr>
<tr><td>LockSetpointEn</td><td>BOOL</td><td>1 = Disables the setpoint of the valve device and lock it into the one defined on the local panel while the local panel mode is Local Panel and the device owner is the Operator. You can switch to the Program mode based on the signals that have been previously described even if the local panel mode is set to Local Panel.

Therefore, user can activate (1) this signal to lock setpoint operation from the monitoring (HMI) system while the valve is controlled from the local panel.</td></tr>
</table>

Outputs

Output Parameter Description

Name	Type	Description
PBLightSig-nal	BOOL	1 = The push buttons on the local panel are fully operational, that is, the local panel is in Local Panel mode (either because the mode is selected with the corresponding switch or because the push buttons are enabled in the event that there is no selector switch available or the *VirtualLPEn* is high with the local panel mode enabled.
LPMode	BOOL	1 = The local panel mode is in Local Panel. The following table shows how the local panel operating mode is determined based on the ModeSignalsEn, VirtualLPEn, PBSignalsEn, LPSignal, CSSignal and ZeroSignal, input signals:

Mode-Signal-sEn	Vir-tualLPEn	PBSigna-lsEn	LPSignal	CSSignal	ZeroSignal	OUTPUT
0	0	0	0	0	0	CS mode
1	0	0	0	0	0	Zero mode
1	0	0	1	0	0	LP mode and PBLightSignal
1	0	0	0	1	0	CS mode
0	1	1	NA	NA	NA	CS mode
0	1	0	NA	NA	NA	CS mode
0	0	1	0	0	0	LP mode and PBLightSignal
1	1	0	1	0	0	LP mode and PBLightSignal
1	1	0	0	0	0	Zero mode
1	1	0	0	0	1	Zero mode
1	1	0	0	1	0	CS mode
1	0	1	0	1	0	CS mode
1	1	1	0	1	0	CS mode
1	0	0	0	0	1	Zero mode

ZeroMode	BOOL	1 = The local panel mode is in Zero. User is not allowed to operate the device from the faceplate and also from DFB. Hence, all the functions in the faceplate will be disabled. Refer to the table included with the description of the LPMode output.
CSMode	BOOL	1 = The local panel mode is in Control System. Refer to the table included with the description of the LPMode output.

Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
MValveDGP_CFG	MValveDGP_CFG_DDT, page 334	Data structure belonging to the control valve is to be controlled from the local panel (for detailed information regarding this structure, refer to the MVALVEDGP DFB, page 329). The structure provides the information needed for the operation of the MVALVEDLPGP DFB. It enables this DFB to control the setpoint for the control valve.
MValveDGP_ST	MValveDGP_ST_DDT, page 334	Data structure belonging to the valve is to be controlled from the local panel (for detailed information regarding this structure, refer to the MVALVEDGP DFB, page 329. The structure provides the information needed for the operation of the MVALVEDLPGP DFB and enables this DFB to control the owner of the valve.
MVal-veDLPGP_ST	MValveDLPGP_ST_DDT	Data structure that is used as an interface with the monitoring (HMI) system.

MValveDLPGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	LPMode	0	BOOL	Refer to the LPMode output pin, page 340.
	ZeroMode	1	BOOL	Refer to the ZeroMode output pin, page 340.
	CSMode	2	BOOL	Refer to the CSMode output pin, page 340.
	OwnerLocked	3	BOOL	Indicates whether the owner change buttons in the monitoring (HMI) system needs to be locked (1) or not (0). Is activated when: • The local panel mode is set to Zero or • It is set to Local Panel, and the LockOwnerEn input signal has been configured as active (1).
	SPLocked	4	BOOL	Indicates whether the set-point change and START buttons in the monitoring (HMI) system needs to be locked (1) or not (0). Is activated when: • The local panel mode is set to Zero or • It is set to Local Panel, and the LockSetpointEn input signal has been configured as active (1).
	ModeSignalsEn	5	BOOL	Refer to the ModeSignalsEn input pin, page 338.
	PBEnabled	6	BOOL	Indicates whether the push buttons of the local panel are enabled (1) or not (0). The signal is activated when at least one of the ModeSignalsEn and PBSignalsEn signal inputs is active.
	VirtualLPEn	7	BOOL	Refer to the VirtualLPEn input pin, page 338.
	LPSignal	8	BOOL	Refer to the LPSignal input pin, page 338.
	ZeroSignal	9	BOOL	Refer to the ZeroSignal input pin, page 338.
	CSSignal	10	BOOL	Refer to the CSSignal inputpin, page 338.
	OffSignal	11	BOOL	Refer to the OffSignal input pin, page 338.
	OpenSignal	12	BOOL	Refer to the OpenSignal input pin, page 338.
	CloseSignal	13	BOOL	Refer to the CloseSignal input pin, page 338.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.

Name	Bit Name	Bit	Type	Description
	LPEnabled	0	BOOL	Read/write access • 1 = Enables Local Panel mode. • 0 = Enables Control System mode.

Public Variables

Public Variable Description

Variable	Type	Description
SC	MValveDLPGP_SC_DDT	Provides the frequently needed data for monitoring and controlling the DFB.

MValveDLPGP_SC_DDT Type

Name	Type	Description
LPMode	BOOL	Read-only access Refer to the LPMode output pin, page 340.
ZeroMode	BOOL	Read-only access Refer to the ZeroMode output pin, page 340.
CSMode	BOOL	Read-only access Refer to the CSMode output pin, page 340.
DisableLP	BOOL	Read/write access 1 = Disables the push buttons on the local panel. Only applicable when a local panel without mode switch configuration is used (ModeSignalsEn = 0).

Supervision

Overview

This section describes the Supervision functions and runtime services that are available for the motorized valve.

Genies

Genie Properties

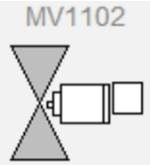
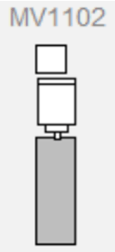
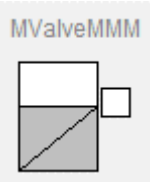
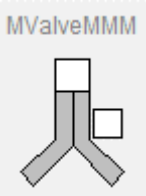
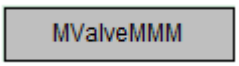
Refer to Genie Properties, page 62

Genie Icons

Refer to Genie Icons, page 63

Representation

The table describes the genies of the `gpl_valvemotorized` library:

Graphic Symbol	Genie Name	Description
	Motor2way_Horiz	Horizontal motorized valve
	Motor2way_Vert	Vertical motorized valve
	Penstock_Vert	Vertical penstock valve
	damper_motorized	Motorized valve - damper
	diverter_motorized	Motorized valve - diverter
	valvemtrzedinstanceid	Motorized valve instance

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

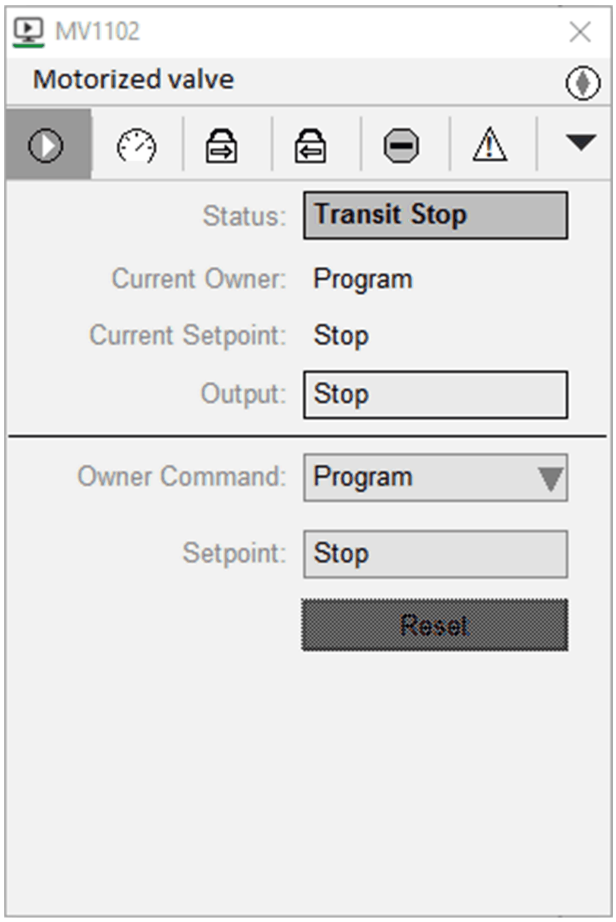
Available Tabs

During operation, clicking a motorized valve genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
  - Event, page 69
- Tabs for optional functions, which appear only if configured:
  - Interlock, page 87
  - Failure, page 78
  - Measures, page 70

Operator Tab

The figure shows an operator tab:



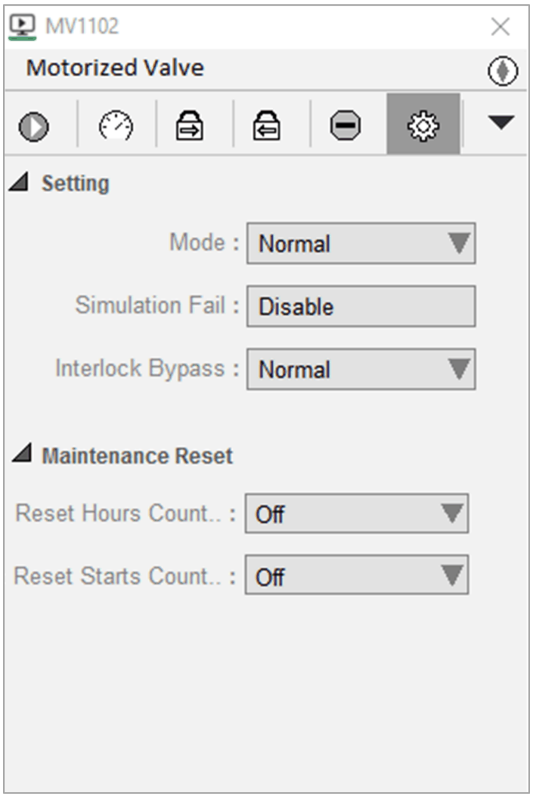
The table describes the control function of an operator tab:

Label	Item Name	Description	Security parameter, page 327
Status	EquipStatus	Equipment status of motorized valve. For more details, refer to EquipStatus calculated variable, page 349.	–
Current Owner	Owner	Current owner of motorized valve. For more details, refer to operating owner, page 58.	
Current Setpoint	CurrentSetpoint	Current setpoint of motorized valve.	

Label	Item Name	Description	Security parameter, page 327
Output	Equipstate	Output status of Motorized Valve. For more details, refer to EquipState calculated variable, page 349.	
OwnerCom-mand	OwnerCommand	Owner command of motorized valve (Program, Operator or Local Panel). **NOTE:** Local panel owner is enabled when the virtual panel is selected.	Set owner
Setpoint	OperatorSet-point	Allows you to change the setpoint, if owner = operator.	Set setpoint
–	Rearm	Allows you to reset manually after the detection of an inoperable device condition or unsuccessful operation. **NOTE:** When the control module is reset, the current setpoint shown in operator tab is effective. Also, when you click **Reset**, a dialog box opens to confirm.	Set rearm

Engineering Tab

The figure shows an engineering tab:



The table describes the control function of engineering tab:

Label	Item name	Description	Security parameter, page 327
Setting			
Mode	Mode	Allows you to view and change the mode. For more details, refer to operating mode, page 59.	Set mode - normal, set mode - override, set mode - simulation, set mode - maintenance, and Set mode - out of service **NOTE:** Each configurable mode has the different security parameters
Simulation Fail	SimulationFailActive	Enable or disable feedback simulation. **NOTE:** The Simulation Fail will only be available	Set mode - simulation

Label	Item name	Description	Security parameter, page 327
		when the parameter `SimulationFail` is created in the citect parameters. To create, go to **Setup** > **Parameters**. Enter the `GPL.Mode` in **Section Name**, `SimulationFail` in **Name** and set the **Value** = 1.	
Interlock Bypass	InterlockBypass	Interlock bypass status Allows you to bypass interlock.	Set bypass conditions, set bypass all interlocks
Maintenance Reset			
Reset Hours Count	MotMntResetHour-sCounter	Reset the hours counter.	Set reset
Reset Starts Count	MotMntResetManeuver-sCounter	Reset the operations counter.	Set reset

Items

Overview

This section describes the variables, equipstatus and equipstate variable, abnormal and action required conditions, alarms, events, and trends items of `motorized valve`.

Variables

The table describes the variable items that are used by Supervision components:

Item Name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`FeedbackOpen`	Open feedback	Open/Not Open	*Bit0*
`FeedbackClosed`	Closed feedback	Closed/Not Closed	*Bit1*
`SetpointOpen`	Open setpoint selected	On/Off	*Bit2*
`SetpointClose`	Close setpoint selected		*Bit3*
`OutputOpen`	Open output status		*Bit4*
`OutputClose`	Close output status		*Bit5*
`ProgramSetpointMode`	Setpoint source for program owner	Remote/Sequence	*Bit6*
`InterlockActive`	Interlock status	Interlocked/Healthy	*Bit7*
`RequireRearm`	Reset required	Required/Not Required	*Bit8*
`OwnerExternal`	External owner status	External/Not External	*Bit9*
`OpenAlarmActive`	Open feedback confirmation detected error	Active/Inactive	*Bit10*
`CloseAlarmActive`	Close feedback confirmation detected error		*Bit11*
`FailActive`	Valve inoperable		*Bit12*

Item Name	Description	Enumeration	Address
ForwardAlarmActive	Forward feedback confirmation detected error		*Bit13*
ReverseAlarmActive	Reverse feedback confirmation detected error		*Bit14*
Bit items derived from StatusWord1.			
NotReady	Not ready to function	–	*Bit0*
OpenSetpointLockAc-tive	Open setpoint lock active		*Bit1*
CloseSetpointLockAc-tive	Close setpoint lock active		*Bit2*
OpenInterlockTripAc-tive	Open interlock trip active		*Bit3*
CloseInterlockTri-pActive	Close interlock trip active		*Bit4*
OpenInterlockActive	Open interlock active		*Bit5*
CloseInterlockActive	Close interlock active		*Bit6*
ForwardRunningFB	Forward running feedback		*Bit8*
ReverseRunningFB	Reverse running feedback		*Bit9*
Bit items derived from ConfigurationWord.			
OwnerSelect	Owner of control module	Operator/Program	*Bit0*
InterlockBypass	Interlock bypass command	Bypassed/Normal	*Bit1*
Rearm	Reset detected error	Active/Inactive	*Bit3*
DirectionSetpoint	Operator direction setpoint	Close/Open	*Bit4*
CommandSetpoint	Operator setpoint	On/Off	*Bit5*
SimulationFailActive	Feedback detected error in simulation mode	–	*Bit12*
Simulation	Simulation mode		*Bit13*
Maintenance	Maintenance mode		*Bit14*
OutOfService	Valve unavailable	Out Of service/In service	*Bit15*
IO Items			
StatusWord	Status word	–	MVALVEDGP_ST.STW
StatusWord1	Status word 1		MVALVEDGP_ST.STW1
ConfigurationWord	Configuration word		MVALVEDGP_ST.CFGW
Calculated Items			
Equipstatus	Current status of the motorized valve For more details, refer to EquipStatus calculated variable, page 349.	–	GPL_MotorizedValveS-tatus
EquipState	Output information For more details, refer to EquipState calculated variable, page 349.		GPL_MotorizedValveE-quipState

Item Name	Description	Enumeration	Address
Owner	Current owner of the motorized valve For more details, refer to calculated variable, page 58.		GPL_OwnerExternalLocal
Mode	Current mode of the motorized valve For more details, refer to calculated variable, page 59.		GPL_ModeMValve
CurrentSetpoint	Current setpoint of the motorized valve		GPL_MotorizedValveCurrentSetpoint
OperatorSetpoint	Operator setpoint of the motorized valve		GPL_MotorizedValveOperatorSetpoint
Abnormal	Shows abnormal condition of the motorized valve For more details, refer to abnormal conditions, page 350.		GPL_Abnormal
ActionRequired	Shows action required condition of the motorized valve For more details, refer to action required conditions, page 350.		GPL_ActionRequired
Optional Items			
Local panel			
Bit items derived from LPStatusWord.			
LocalPanelMode	Control from field	Enabled/Disabled	*Bit0*
ZeroMode	valve control disabled		*Bit1*
ControlSystemMode	Control from remote		*Bit2*
LockedOwner	Owner change disabled from faceplate	Locked/Unlocked	*Bit3*
LockedSetpoint	Setpoint change disabled from faceplate		*Bit4*
LocalPanelEnabled	Control from field	Enabled/Disabled	*Bit5*
PushButtonEnabled	Local panel push buttons		*Bit6*
VirtualLocalPanelEnabled	Local panel enable from control		*Bit7*
LocalPanelSelected	Control from field	Selected/Not Selected	*Bit8*
ZeroSelected	Control disabled		*Bit9*
ControlSystemSelected	Control from remote		*Bit10*
PushButtonStop	Stop push button signal	On/Off	*Bit11*
PushButtonOpen	Open push button signal		*Bit12*
PushButtonClose	Close push button signal		*Bit13*
Bit items derived from LPConfigurationWord.			

Item Name	Description	Enumeration	Address
VirtualSwitch	Local panel selection from faceplate	Local/Control System	*Bit0*
IO Items			
LPStatusWord	Local panel status word	–	MVALVEDLPGP_ST.STW
LPConfigurationWord	Local panel configuration word		MVALVEDLPGP_ST.CFGW
Failures	Refer to the topic describing the variable items of failures, page 61.		
Interlocks	Refer to the topic describing the variable items of interlocks, page 60.		
Measures	Refer to the topic describing the variable items of measures, page 59.		

EquipStatus Calculated Variable for MValveD

The table describes the *EquipStatus* calculated variable:

Equipment Status	Conditions	Enumeration
–	(Open x Closed) + (FwdRun x RevRun)	Unknown
Valve Close	Not Open x Closed	Closed
Valve Open	Open x Not Closed	Open
Valve Transition	FwdCmd x Not Open	Opening
	RevCmd x Not Close	Closing
	Not Open x Not Closed x Not FwdCmd x Not RevCmd x (FwdRun + RevRun)	Stopping
Valve Stopped	Not Open x Not Closed x Not FwdCmd x Not RevCmd x Not FwdRun x Not RevRun	Stopped

EquipStatus Calculated Variable for MValveD with Motor

The table describes the *EquipStatus* calculated variable:

Equipment Status	Conditions	Enumeration
–	FwdRun x RevRun	Unknown
Valve Transition	(Not FwdCmd x Not RevCmd) x ((FwdRun x Not RevRun) + (Not FwdRun x RevRun))	Stopping
Valve Stopped	Not FwdCmd x Not RevCmd x Not FwdRun x Not RevRun	Stopped
Valve Transition	FwdRun x Not RevRun	Running Forward
	Not FwdRun x RevRun	Running Reverse
	FwdCmd x Not FwdRun	Starting Forward
	RevCmd x Not RevCmd	Starting Reverse

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	Conditions	Enumeration
Passive with Out of Service	OOS	Stop
Active in Maintenance	Not OOS x Maint x OpenCmd	Open
Passive in Maintenance	Not OOS x Maint x CloseCmd	Close
Passive with Out of Service	Not OOS x Maint x Not OpenCmd x Not CloseCmd	Stop

Equipment State	Conditions	Enumeration
Stop	Not `OOS` x Not `Maint` x `Fail`	`Failed`
Active with Interlock	Not `OOS` x Not `Maint` x Not `Fail` x `OpenCmd` x `OpenILCK`	Open Interlocked
Active	Not `OOS` x Not `Maint` x Not `Fail` x `OpenCmd` x Not `OpenILCK`	Open
Passive with Interlock	Not `OOS` x Not `Maint` x Not `Fail` x `CloseCmd` x `CloseILCK`	Close Interlocked
Active to Passive	Not `OOS` x Not `Maint` x Not `Fail` x `CloseCmd` x Not `CloseILCK`	Close
Passive with Interlock	Not `OOS` x Not `Maint` x Not `Fail` x Not `OpenCmd` x Not `CloseCmd` x (`OpenILCK` + `CloseILCK`)	Interlocked
Passive	Not `OOS` x Not `Maint` x Not `Fail` x Not `OpenCmd` x Not `CloseCmd` x Not `OpenILCK` x Not `CloseILCK`	Stop

Abnormal Conditions

The table describes the abnormal conditions of motorized valve:

Item Name	Description	Address
Bit items derived from Abnormal variable word.		
`AbnormalOwner`	Abnormal owner status appears, if the current owner is not a normal owner which configured at the instance level.	*Bit0*
`AbnormalMode`	Abnormal mode status appears, if the current operating mode is not normal mode.	*Bit1*
`AbnormalInterlockBy-pass`	Abnormal interlock bypass condition appears, if the Interlock bypass is enabled.	*Bit2*
`AbnormalOperatorTab`	Abnormal status of operator tab.	*Bit8*
`AbnormalFailTab`	Abnormal status of **Failures** tab.	*Bit10*
`AbnormalEngineeringTab`	Abnormal status of engineering tab.	*Bit12*
`AbnormalDirection1In-terlockTab`	Abnormal status of direction 1 interlock tab.	*Bit13*
`AbnormalDirection2In-terlockTab`	Abnormal status of direction 2 interlock tab.	*Bit14*

Action Required Conditions

The table describes the action required conditions of motorized valve:

Item Name	Description	Address
Bit items derived from ActionRequired variable word.		
`ActionOperatorTab`	Action required status of operator tab.	*Bit8*
`ActionFailTab`	Action required status of **Failures** tab.	*Bit10*
`ActionDirection1In-terlockTab`	Action status of direction 1 interlock tab.	*Bit13*
`ActionDirection2In-terlockTab`	Action status of direction 2 interlock tab.	*Bit14*

Alarms

The table describes the advanced alarm items that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
VlvMntOperationHoursAlm	Valve running hours Counter overflow	GPL_BitValueCheck (StatusWord, 0)
VlvMntManeuversNumbersAlm	Valve operations counter overflow	GPL_BitValueCheck (StatusWord, 1)
MotMntOperationHoursAlm	Motor running hours Counter overflow	GPL_BitValueCheck (StatusWord, 0)
MotMntManeuversNumbersAlm	Motor operations counter overflow	GPL_BitValueCheck (StatusWord, 1)
AlarmValue	Valve confirmation detected error	GPL_BitValueCheck (StatusWord, 10,11)
AlarmMotor	Motor confirmation detected error	GPL_BitValueCheck (StatusWord, 13,14)
Fail	Inoperable motorized valve	GPL_BitValueCheck (StatusWord1, 11)

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item Name	Description
StateAlarm	All changes in the motorized valve status and output.
ModeAlarm	All changes in the mode of operation of the equipment.
ActionAlarm	All the operator actions in the supervision.

Trends

No trend item is used by Supervision components for motorized control valve.

Control Valve

What's in This Chapter

Template 352
Control 356
Optional Service - Local Panel (CVALVELPGP) 362
Supervision 367

Overview

This chapter describes the functionality of template, control services and Supervision functions of the control valve.

Template

Overview

This section describes the functionality, parameters, and interfaces of the $ControlValveGP template.

Description

General Description

The `$ControlValveGP` process control module template manages control valves with optional position feedback (position and/or limit switches).

Function Description

The main functions of the control valve template are described in the following table:

Function	Description
Owner	The object manages the control system level which is the owner (Operator or Program). As a result, it is responsible for setting the set-point and activation for the control.
Interlocking	The template gives a command to the device to move to the defined position in case of an active interlock. An interlock bypass function is available.
Local and remote set-point	Enables to control the template with a set-point that is determined through a set-point selector (local or remote) in program owner. The local set-point is assigned from the sequential control while the remote set-point is assigned to the template from the logic implemented in the continuous/ trigger based control.
Mode selection	The object can be switched to different modes of operation like Simulation, Maintenance, Out Of Service and Override.
Tracking	The template activates monitoring for tracking the actual position in relation to the target position.
Detection of not confirmed operation	If position is not confirmed within a configured time after the output command is activated, template detects not confirmed operation.

Optional Services

Available optional services are:

- Interlock
- Local Panel

Parameters

Parameters

The `$ControlValveGP` process control module template manages control valves with optional position feedback (position and/or limit switches).

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	`Valve in case of channel failure`	`Float`	0.0	Default value in engineering units to be used for the output signal when the channel is not operational.
	Control valve setpoint when interlocked.			Setpoint in engineering units to be used for the output signal when the channel is not interlocked.
Setpoint	Force remote setpoint	`Boolean`	False	Enables the remote control operation for motor.
Range	High for the channel input signal	`Short`	2047	–
	Low for the channel input signal		0	–
	High for the channel output signal, in raw data		2047	–
	Low for the channel output signal, in raw data		0	–
	Open valve threshold	`Float`	95.0	Minimum position to consider valve as open.
	Closed valve threshold	`Float`	5.0	Maximum position to consider valve as closed.
Time	Maximum permissible deviation between current setpoint and actual position	`Float`	5.0	–
	Maximum permissible time during which there can be a follow-up	`Duration`	00:00:05	Maximum allowed time for input to follow output.
	`Maximum time to maintain warning signal`		00:00:05	Minimum time to maintain alarms.

Inputs (Digital Input)

The table describes the digital input:

Name	Description
`OpenPosition`	It allows you to connect the open feedback of the valve.
`ClosePosition`	It allows you to connect the closed feedback of the valve.
`PV`	It allows to connect analog input to track the position of the valve.

For more details, refer to the topic describing the parameters of digital inputs, page 44.

Outputs (Digital Output)

The table describes the digital output:

Name	Description
`OP`	It allows you to connect the digital output signal of the valve to external device.

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Local Panel (Logic)

For more details, refer to the topic describing the parameters of logic, page 45.

Local Panel (Digital Input)

The table describes the digital input:

Name	Description
LPMode	Local panel selection signal from mode selector switch.
ZeroMode	Zero/Off selection signal from mode selector switch.
CSMode	Control selection signal from selector switch.
OpenPB	Start signal from local panel.
ClosePB	Stop signal from local panel.

For more details, refer to the topic describing the parameters of digital inputs, page 44.

Local Panel (Digital output)

The table describes the digital output:

Name	Description
PBLight	Output signal to indicate that push buttons are operational.

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Operation	Normal owner	`Enum`	Program and Cascade (5)	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	Set owner	`Enum`	Operator (1)	Privilege to change owner. This parameter is used in Operator Tab, page 370.
	Set setpoint		Operator (1)	Privilege to change setpoint. This parameter is used in Operator Tab, page 370.
	Set mode - override		Supervisor (2)	Privilege to change override mode.This parameter is used in Engineering Tab, page 371.
	Set mode - simulation		Engineer (5)	Privilege to change simulation mode. This parameter is used in Engineering Tab, page 371.
	Set mode - maintenance		Maintenance (3)	Privilege to change maintenance mode. This parameter is used in Engineering Tab, page 371.
	Set mode - out of service		Supervisor (2)	Privilege to change out of service mode. This parameter is used in Engineering Tab, page 371.
	Set reset		Operator (Confirmed) (10)	Privilege to reset. This parameter is used in Operator Tab, page 370.
	Set alarm acknowledge		Operator (1)	Privilege to acknowledge alarm. This parameter is used in Alarm Tab, page 67.
	Set bypass conditions		Supervisor (2)	Privilege to set bypass conditions mode. This parameter is used in Engineering Tab, page 371.
	Set bypass all interlocks		Supervisor (2)	Privilege to bypass all interlocks. This parameter is used in Engineering Tab, page 371.
Trend	–	–	–	For more details, refer to the topic describing the parameters of trend, page 47.
Events	Enable log	`Boolean`	False	Enable event logging for asset status.
Historize	Output	`Boolean`	False	–.
	Local setpoint			It allows to historize the change local setpoint.
	Present value			It allows to historize the change present value.

Asset and Display Parameters

For more details, refer to the topic describing the parameters of asset and display, page 43.

Alarm Parameters

For more details, refer to the topic describing the parameters of alarm, page 45.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Local Panel

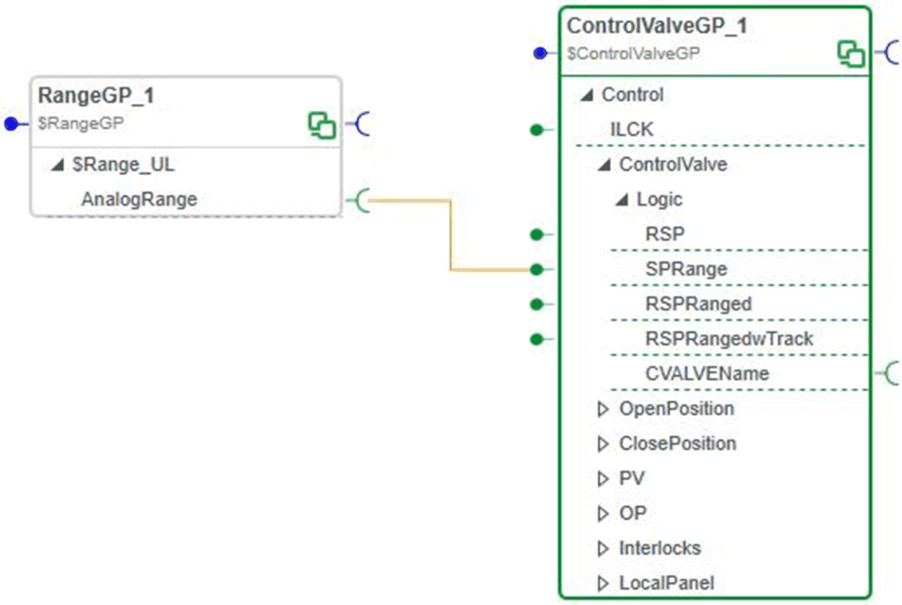
For more details, refer to the topic describing the parameters of local panel, page 44.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$ControlValveGP` (also applicable for `$ControlValveCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



The following table lists the link interface that creates link symbol on faceplate:

Interface name	Runtime parameter
`RSP`	SetpointNavigate
`RSPRanged`	
`RSPRangedwTrack`	

For more details, refer to Supervision chapter > Faceplate > Navigation Link of General Purpose Library User Guide.

Refer to common control module template interface, page 47 for more details on common interfaces.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

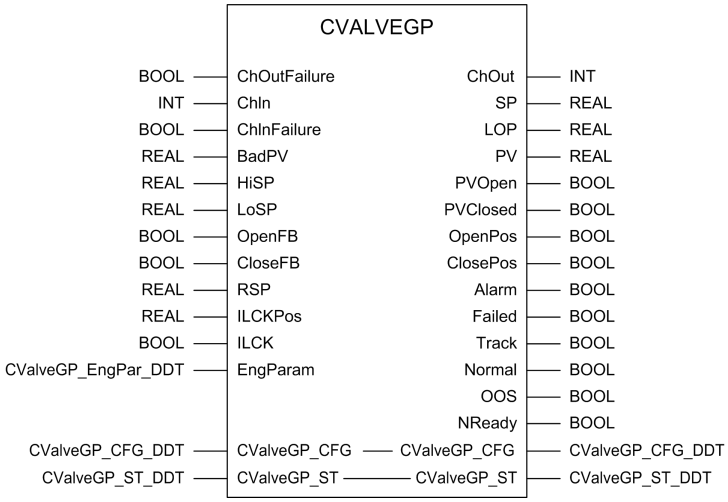
Overview

This section describes the `CVALVEGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Description Table

Name	Type	Description
ChOutFailure	BOOL	1 = Indicates to the DFB the output channel associated with the ChOut output is operational.
ChIn	INT	Input signal that indicates the position of the valve. Normally corresponds to a value in raw data coming from the input/output peripherals. Connecting a variable or value to this DFB pin enables analog position feedback.
ChInFailure	BOOL	1 = Indicates to the DFB the input channel associated with the ChIn input is operational as long as position feedback is enabled (refer to the ChIn input).
BadPV	REAL	Value in engineering units (PV output signal) needs to be used when the channel is non-operational.
HiSP	REAL	High range for the set-point (SP) corresponding to the maximum value (configured in the HiRawChOut input) of the ChOut output in engineering units.
LoSP	REAL	Low range for the set-point (SP) in engineering units corresponding to the minimum value (configured in the LoRawChOut input) of the ChOut output in engineering units.
OpenFB	BOOL	1 = Indicates that the open valve position is reached. Used to connect to the digital input of the open-valve limit switch. Disabled internally by the DFB if no variable or value is connected.
CloseFB	BOOL	1 = indicates that the closed valve position is reached. Used to connect to the digital input of the closed-valve limit switch. Disabled internally by the DFB if no variable or value is connected.
RSP	REAL	Remote set-point by the continuous control for control PID controller output.
ILCKPos	REAL	Position to be considered during interlock.

Name	Type	Description
ILCK	BOOL	1 = Interlocks the device at the defined position.
EngParam	CVal-veGP_EngPar_DDT	Engineering parameters

CValveGP_EngPar_DDT Type

Name	Type	Description
HiRawChOut	INT	High range for the ChOut output signal in raw data.
LoRawChOut	INT	Low range for the ChOut output signal in raw data.
HiRawChIn	INT	High range for the ChIn input signal.
LoRawChIn	INT	Low range for the ChIn input signal.
MaxDev	REAL	Enables to set the maximum permissible deviation between the current set-point and the actual position. If the difference is greater than the value of this parameter for a certain period (refer to the TMaxDev engineering parameter), the follow-up detected fault alarm is activated.
TMaxDev	TIME	Maximum permissible time during which there can be a follow-up detected fault. The follow-up detected fault is generated if the difference between the set-point and the actual position is greater than the value entered in the MaxDev engineering parameter. You can deactivate follow-up alarm monitoring by setting this parameter to 0 (T#0s).
ScanTime	TIME	Enables to configure the time during which the follow-up alarm signal is kept active. This helps the monitoring subsystem to acquire the data in case of brief alarms detected.
HiThreshold	REAL	Enables to configure the open-valve threshold for the analog position measurement as a percentage, that is, the threshold after which it is considered that the valve is completely opened.
LoThreshold	REAL	Enables to configure the closed-valve threshold for the analog position measurement as a percentage, that is, the threshold after which it is considered that the valve is completely closed.

Outputs

Output Parameter Description

Name	Type	Description	
ChOut	INT	Calculated value of the output in raw data is used to position the valve. The ChOut calculation is shown in the following table.	
		OP	**ChOut** is calculated as:
		<= LoSP	LoRaw
		> LoSP and < HiSP	Refer to the following linear scaling formula.
		>= HiSP	HiRaw
		NOTE: In simulation mode, ChOut holds the last value.	
SP	REAL	Current valve set-point. The SP calculation is shown in the following table:	
		Owner (OFF: Program, ON: Operator)	**Rem** / **SP** is calculated as:
		OFF	OFF / SC.LSP

Name	Type	Description		
		OFF	ON	RSP
		ON	-	CValveGP_CFG.LSP
LOP	REAL	Valve positioning output in engineering units. The OP calculation is shown in the following table:		
		SC.ILCKD	**OP** is calculated as:	
		ON	ILCKPos	
		OFF	**SP**	
PV	REAL	Current valve position in engineering units.		
PVOpen	BOOL	1 = Indicates the open-valve position has been reached. This is verified by checking whether or not the analog input falls within the threshold specified by means of the HiThreshold input.		
PVClosed	BOOL	1 = Indicates the closed-valve position has been reached. This is verified by checking whether or not the analog input falls within the threshold specified by means of the LoThreshold input.		
OpenPos	BOOL	1 = Indicates the open-valve position has been reached. If OpenPos is not operational, PVOpen is used as a basis for the indication instead.		
ClosePos	BOOL	1 = Indicates the closed-valve position has been reached. If ClosePos is not operational, PVClosed is used as a basis for the indication instead.		
Alarm	BOOL	1 = Indicates a follow-up detected fault (1) has occurred. The analog position signal is monitored. Refer to the MaxDev and TMaxDev engineering parameters, page 357.		
Failed	BOOL	1 = Indicates an inoperable device (1) in the either of the following cases: • when the OpenFB and CloseFB inputs are active simultaneously • when the limit switches do not correspond to the maximum and minimum values that are determined based on the measurement of current position.		
Track	BOOL	Refer to Track, page 400 output pin of AOutputGP DFB.		
Normal	BOOL	Indicates the motor is operating in normal mode.		
OOS	BOOL	1 = Indicates the valve is out of service.		
NReady	BOOL	Indicates valve is not ready for operation.		

Linear Scaling Formula:

$$\left(\frac{(HIRAW - LORAW)}{(HISP - LOSP)} \times SP\right) + LORAW - \left(\frac{(HIRAW - LORAW)}{(HISP - LOSP)} \times LOSP\right)$$

Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
CValveGP_CFG	CValveGP_CFG_DDT	Provides the data necessary to configure the DFB (normally from the monitoring subsystem).
CValveGP_ST	CValveGP_ST_DDT	Provides the data necessary to monitor the DFB status.

CValveGP_CFG_DDT Type

Name	Type	Description
LSP	REAL	This variable indicates the current set-point. It is calculated internally in the DFB if the Program is the owner. It is set to match the SP output. If the Operator is the owner, you can modify it from the monitoring subsystem.

CValveGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
PV	–	–	REAL	Read-only access Refer to the PV output pin, page 358.
LOP	–	–	REAL	Read-only access Refer to the LOP output pin, page 358.
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	ChInFailure	0	BOOL	Refer to the ChInFailure input pin, page 357.
	ILCK	1	BOOL	Refer to the ILCK input pin, page 357.
	Rem	2	BOOL	Refer to the Rem, page 360.
	ChOutFailure	4	BOOL	Refer to the ChOutFailure input pin, page 357.
	PVOpen	6	BOOL	Refer to the PVOpen output pin, page 358.
	PVClosed	7	BOOL	Refer to the PVClosed output pin, page 358.
	NReady	8	BOOL	Refer to the NReady output pin, page 358.
	Failed	9	BOOL	Refer to the Failed output pin, page 358.
	Alarm	10	BOOL	Refer to the Alarm output pin, page 358.
	ClosePos	11	BOOL	Refer to the ClosePos output pin, page 358.
	OpenPos	12	BOOL	Refer to the OpenPos output pin, page 358.
	CloseLimitEn	13	BOOL	1 = Closed limit switch signal connected.
	OpenLimitEn	14	BOOL	1 = Open limit switch signal connected.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
	Owner	0	BOOL	Read/write access Refer to the Owner, page 360.
	ILCKBP	1	BOOL	Read/write access 1 = Allows the interlock to be bypassed.
	Override	2	BOOL	Refer to the Override, page 360.
	SimFBFail	12	BOOL	1 = Feedback will not simulate in simulation mode.
	Sim	13	BOOL	Read/write access Refer to the Sim, page 360.
	Maint	14	BOOL	Refer to the Maint, page 360.
	OOS	15	BOOL	Refer to the OOS output pin, page 358.

Public Variables

Public Variable Description

Variable	Type	Description
SC	CValveGP_SC_DDT	Provides the frequently needed data to monitor the DFB status.

CValveGP_SC_DDT Type

Name	Type	Description
LSP	REAL	Read/write access Enables the sequential control to assign the local set-point if the owner is the Program (Owner input/output is set to 0), and the selected set-point is the local one (SC.Rem public variable is set to 0). Otherwise, the current set-point (SP output) is copied continuously to this variable.
LOP	REAL	Read-only access Refer to the LOP output pin, page 358.
PV	BOOL	Read-only access Refer to the PV output pin, page 358.
Rem	BOOL	Read/write access 1 = Enables the DFB to be configured to remote set-point RSP 0 = Enables the DFB to be configured to local set-point LSP.
PVOpen	BOOL	Read-only access Refer to the PVOpen output pin, page 358.
PVClosed	BOOL	Read-only access Refer to the PVClosed output pin, page 358.
ClosePos	BOOL	Read-only access Refer to the ClosePos output pin, page 358.
OpenPos	BOOL	Read-only access Refer to the OpenPos output pin, page 358.
BadSTIn	BOOL	Read-only access Refer to the ChInFailure input pin, page 357.
BadSTOut	BOOL	Read-only access Refer to the ChOutFailure input pin, page 357.
Alarm	BOOL	Read-only access Refer to the Alarm output pin, page 358.
Failed	BOOL	Read-only access Refer to the Failed output pin, page 358.
ILCKD	BOOL	Read-only access Indicates whether the motor is interlocked. The signal evaluation depending on the ILCK input and the CValveGP_ST.CFGW.ILCKBP input/output is shown in the following table: **ILCK** / **ILCKBP** / **SC.ILCKD.LSP** OFF / - / OFF ON / OFF / ON ON / ON / OFF
Owner	BOOL	Read/write access Enables to configure whether the set-point is set by the Program (0) or by the Operator (1).
Normal	BOOL	Read-only access Refer to the Normal output pin, page 358.
Override	BOOL	Read/write access

Name	Type	Description
		Enables to set the valve to `Override` mode so that the actual position signals (`PV` input) are ignored. It is simulated that the valve is in the desired position (according to the `LOP` output pin) and is functional.
`Sim`	`BOOL`	Read/write access Enables to set the valve to simulation mode so that the actual position signals (`PV` input) are ignored. It is simulated that the valve is in the desired position (according to the `LOP` output pin) and the `ChOut` output pin holds the last value.
`Maint`	`BOOL`	Read-only access 1 = Maintenance mode active. Allows the operation of the valve for maintaining and testing. Interlock conditions are bypassed.
`OOS`	`BOOL`	Read/write access Refer to the `OOS` output pin, page 358.
`NReady`	`BOOL`	Read-only access Refer to the `NReady` output pin, page 358.

Optional Service - Local Panel (CVALVELPGP)

Overview

This chapter describes the optional services - local panel (CVALVELPGP) DFB.

Description

General

The main objective of the `CVALVELPGP` DFB is to manage a local panel that controls a control valve with position feedback. This valve is implemented with a `CVALVEGP` function block and with signals that are wired to the controller so that the latter defines the target position for the device.

Function Description

The main functions of the module are described in the following table:

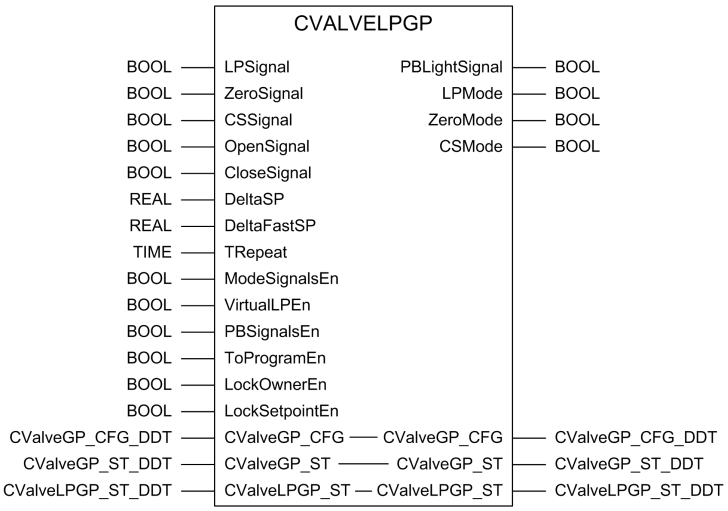
Function	Description
Mode Switch	Optionally manages the signals coming from an operating mode switch in a local panel with the following configuration: Local - Zero - Control System. The Zero mode signal is optional. However the user can enable Local/Control System mode from the faceplate also, when *VirtualLPEn* input pin signal is high and *ModeSignalsEn* signal is low.
Push buttons	The DFB manages up to two signals coming from OPEN and CLOSE push buttons, giving the CLOSE push button higher priority.
Owner Management	The DFB enables to configure whether the Program needs to be the valve owner or not, after switching to the Control System mode again.
Owner Locking	The DFB enables to configure whether the monitoring (HMI) system needs to block access to the drop-down list or not, it disables the access to change the owner (Operator/Program) while the valve is controlled from the Local Panel.

Function	Description
Set-point Lock	The DFB enables to configure whether the monitoring (HMI) system needs to block access to the drop-down list or not. It disables to change the set-point while the valve is controlled from the Local Panel.
Push button Enabling/Disabling	The push button signals from the Local Panel can be enabled/disabled through the DFB configuration (input pin) and/or from control sequences.
Enabled Panel Signaling	The DFB provides a signal that can be used to illuminate a light source on the Local Panel to indicate when the push buttons are enabled for operation.
Virtual Panel Enabling/Disabling	This signal enables the operator to select the Local / Control System mode of operation from the HMI so that the push button signals are enabled for operation.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Name	Type	Description
`LPSignal`	`BOOL`	1 = The local panel mode switch is in the Local Panel position. Refer to the table included with the description of the `LPMode` output pin, page 365.
`ZeroSignal`	`BOOL`	1 = The local panel mode switch is in the zero position.This signal is optional even if a mode switch is available on the local panel. When this signal is a logical high, user is not allowed to operate the device from the faceplate and also from the DFB. Hence, all the functions in the faceplate will be disabled. Refer to the table included with the description of the `LPMode` output pin, page 365.
`CSSignal`	`BOOL`	1 = The local panel mode switch is in the Control System position. Refer to the table included with the description of the `LPMode` output pin, page 365.
`OpenSignal`	`BOOL`	1 = Indicates to the DFB the OPEN push button on the local panel is pressed.
`CloseSignal`	`BOOL`	1 = Indicates to the DFB the CLOSE push button on the local panel is pressed.
`DeltaSP`	`REAL`	Absolute set-point increase or decrease value when the OPEN or CLOSE push buttons are pressed.
`DeltaFastSP`	`REAL`	Absolute set-point increase or decrease value when the OPEN and CLOSE push buttons are being pressed.

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>TRepeat</td><td>TIME</td><td>Indicates the maximum time that needs to elapse before the system interprets that a command has been repeated when the push button is pressed. This time also determines when a DeltaFastSP increase or decrease has to be applied when the OPEN or CLOSE push button is pressed.</td></tr>
<tr><td>ModeSignalsEn</td><td>BOOL</td><td>1 = Enables the use of the Local Panel/Zero/ Control System (or Local Panel - Control System) mode switch on the local panel. Normally configured in development phase based on the characteristics of the local panel being used. If the mode switch is enabled, the LPSignal, ZeroSignal, and CSSignal inputs are considered to determine the operating modes of the local panel. Refer to the table included with the description of the LPMode output pin, page 365.
NOTE: ModeSignalsEn input pin has higher priority than the VirtualLPEn input pin signal.</td></tr>
<tr><td>VirtualLPEn</td><td>BOOL</td><td>1 = This signal enables the operator to select the Local/Control System mode of operation from the HMI, so that the push button signals are enabled for operation.
NOTE: When the owner is program, the Local/Control System mode selection drop-down list will be visible but disabled for operation however, when the owner is operator the Local Panel drop-down list will be accessible for operation.
Refer to the table included with the description of the LPMode output pin, page 365.</td></tr>
<tr><td>PBSignalsEn</td><td>BOOL</td><td>1 = Enables to accept the OPEN and CLOSE push buttons from the local panel. This signal is not applicable if the mode switch is enabled (refer to the ModeSignalsEn input). This means that the push button signals are considered in the Local Panel mode.

When the mode switch is disabled (ModeSignalsEn = 0), PBSignalsEn signal enables/disables the push button signals from control system itself based on relevant process conditions. SC.DisableLP public variable is also considered for determining whether the push buttons are enabled or not when LPMode pin signal is high as shown in the following table.
<table>
<tr><th>VirtualLPEn</th><th>ModeSignalsEn</th><th>PBSignal-sEn</th><th>SC.DisableLP</th><th>Push buttons enabled?</th></tr>
<tr><td>OFF</td><td>OFF</td><td>OFF</td><td>-</td><td>NO</td></tr>
<tr><td>OFF</td><td>OFF</td><td>-</td><td>ON</td><td>NO</td></tr>
<tr><td>OFF</td><td>OFF</td><td>ON</td><td>OFF</td><td>YES</td></tr>
<tr><td>OFF</td><td>ON</td><td>-</td><td>-</td><td>YES</td></tr>
<tr><td>ON</td><td>OFF</td><td>OFF</td><td>-</td><td>YES</td></tr>
</table>
Therefore, user can use this signal to enable/disable the push buttons on a simple local panel that does not feature a mode switch (Local Panel/Zero/Control System) based on the relevant process conditions.</td></tr>
<tr><td>ToProgramEn</td><td>BOOL</td><td>1 = Change to program mode when control is switched to control system mode and VirtualLPEn = 0.</td></tr>
<tr><td>LockOwnerEn</td><td>BOOL</td><td>1 = Disables the access of owner of the valve remaining as Operator while the local panel mode is Local Panel.

The following table shows how the valve owner is evaluated based on the ToProgramEn and LockOwnerEn inputs and operating mode of the local panel:
<table>
<tr><th>Local Panel Mode</th><th>ToProgra-mEn</th><th>LockOwne-rEn</th><th>Owner</th></tr>
<tr><td>Zero</td><td>–</td><td>–</td><td>Operator</td></tr>
<tr><td>Switch to Local Panel</td><td>–</td><td>–</td><td>Operator</td></tr>
<tr><td>Local Panel</td><td>–</td><td>ON</td><td>Operator</td></tr>
<tr><td>Switch to Control System</td><td>ON</td><td>–</td><td>Program</td></tr>
</table>
In remaining cases, owner of the valve ceases to be defined from the CVALVELPGP DFB, and it enables to change it from the monitoring (HMI) system with the command word in the CValveGP_ST.CFGW input/output of the corresponding CVALVEGP DFB.

In any case, consider that the owner of valve that had been set before switching to Local Panel mode is not memorized.</td></tr>
<tr><td>LockSetpointEn</td><td>BOOL</td><td>1 = Disables the setpoint of the valve device and lock it into the one defined on the local panel while the local panel mode is Local Panel and the device owner is the Operator. You can switch to the Program mode based on the signals that have been previously described even if the local panel mode is set to Local Panel.

Therefore, user can activate (1) this signal to lock setpoint operation from the monitoring (HMI) system while the valve is controlled from the local panel.</td></tr>
</table>

Outputs

Output Parameter Description

Name	Type	Description
PBLightSig-nal	BOOL	1 = The push buttons on the local panel are fully operational, that is, the local panel is in Local Panel mode (either because the mode is selected with the corresponding switch or because the push buttons are enabled in the event that there is no selector switch available or the *VirtualLPEn* is high with the local panel mode enabled).
LPMode	BOOL	1 = The local panel mode is in Local Panel. The following table shows how the local panel operating mode is determined based on the ModeSignalsEn, VirtualLPEn, PBSignalsEn, LPSignal, CSSignal and ZeroSignal, input signals:

Mode-Signal-sEn	Vir-tualLPEn	PBSigna-lsEn	LPSignal	CSSignal	ZeroSignal	OUTPUT
0	0	0	0	0	0	CS mode
1	0	0	0	0	0	Zero mode
1	0	0	1	0	0	LP mode and PBLightSignal
1	0	0	0	1	0	CS mode
0	1	1	NA	NA	NA	CS mode
0	1	0	NA	NA	NA	CS mode
0	0	1	0	0	0	LP mode and PBLightSignal
1	1	0	1	0	0	LP mode and PBLightSignal
1	1	0	0	0	0	Zero mode
1	1	0	0	0	1	Zero mode
1	1	0	0	1	0	CS mode
1	0	1	0	1	0	CS mode
1	1	1	0	1	0	CS mode
1	0	0	0	0	1	Zero mode

ZeroMode	BOOL	1 = The local panel mode is in Zero. User is not allowed to operate the device from the faceplate and also from DFB. Hence, all the functions in the faceplate will be disabled. Refer to the table included with the description of the LPMode output.
CSMode	BOOL	1 = The local panel mode is in Control System. Refer to the table included with the description of the LPMode output.

Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
CValveGP_CFG	CValveGP_CFG_DDT, page 359	Data structure belonging to the control valve is to be controlled from the local panel (for detailed information regarding this structure, refer to the CVALVEGP DFB, page 357). The structure provides the information needed for the operation of the CVALVELPGP DFB. It enables this DFB to control the setpoint for the control valve.
CValveGP_ST	CValveGP_ST_DDT, page 359	Data structure belonging to the valve is to be controlled from the local panel (for detailed information regarding this structure, refer to the CVALVEGP DFB, page 357. The structure provides the information needed for the operation of the CVALVELPGP DFB and enables this DFB to control the owner of the valve.
CVal-veLPGP_ST	CValveLPGP_ST_DDT	Data structure that is used as an interface with the monitoring (HMI) system.

CValveLPGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	LPMode	0	BOOL	Refer to the LPMode output pin, page 365.
	ZeroMode	1	BOOL	Refer to the ZeroMode output pin, page 365.
	CSMode	2	BOOL	Refer to the CSMode output pin, page 365.
	OwnerLocked	3	BOOL	Indicates whether the owner change buttons in the monitoring (HMI) system needs to be locked (1) or not (0). Is activated when: • The local panel mode is set to Zero or • It is set to Local Panel, and the LockOwnerEn input signal has been configured as active (1).
	SPLocked	4	BOOL	Indicates whether the set-point change and START buttons in the monitoring (HMI) system needs to be locked (1) or not (0). Is activated when: • The local panel mode is set to Zero or • It is set to Local Panel, and the LockSetpointEn input signal has been configured as active (1).
	ModeSignalsEn	5	BOOL	Refer to the ModeSignalsEn input pin, page 363.
	PBEnabled	6	BOOL	Indicates whether the push buttons of the local panel are enabled (1) or not (0). The signal is activated when at least one of the ModeSignalsEn and PBSignalsEn signal inputs is active.
	VirtualLPEn	7	BOOL	Refer to the VirtualLPEn input pin, page 363.
	LPSignal	8	BOOL	Refer to the LPSignal input pin, page 363.
	ZeroSignal	9	BOOL	Refer to the ZeroSignal input pin, page 363.
	CSSignal	10	BOOL	Refer to the CSSignal input pin, page 363.
	OpenSignal	11	BOOL	Refer to the OpenSignal input pin, page 363.
	CloseSignal	12	BOOL	Refer to the CloseSignal input pin, page 363.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.

Name	Bit Name	Bit	Type	Description
	LPEnabled	0	BOOL	Read/write access • 1 = Enables Local Panel mode. • 0 = Enables Control System mode.

Public Variables

Public Variable Description

Variable	Type	Description
SC	CValveLPGP_SC_DDT	Provides the frequently needed data for monitoring and controlling the DFB.

CValveLPGP_SC_DDT Type

Name	Type	Description
LPMode	BOOL	Read-only access Refer to the LPMode output pin, page 365.
ZeroMode	BOOL	Read-only access Refer to the ZeroMode output pin, page 365.
CSMode	BOOL	Read-only access Refer to the CSMode output pin, page 365.
DisableLP	BOOL	Read/write access 1 = Disables the push buttons on the local panel. Only applicable when a local panel without mode switch configuration is used (ModeSignalsEn = 0).

Supervision

Overview

This section describes the Supervision functions and runtime services that are available for the control valve.

Genies

Genie Properties

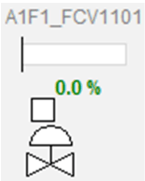
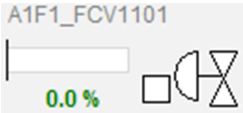
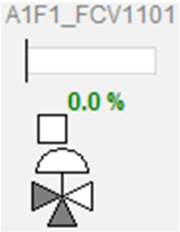
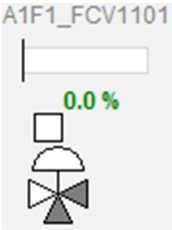
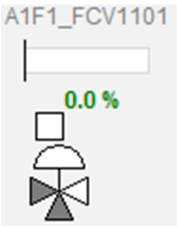
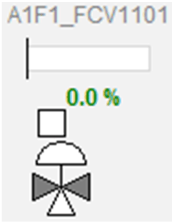
Refer to Genie Properties, page 62.

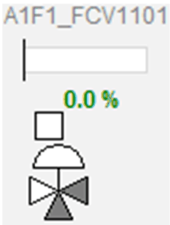
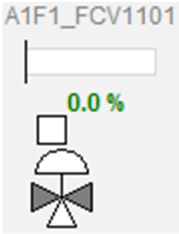
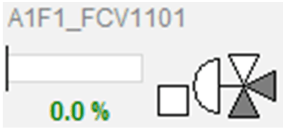
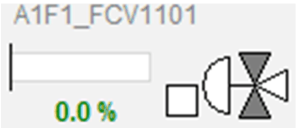
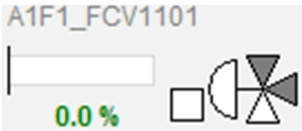
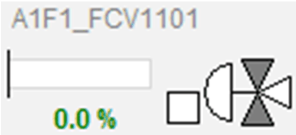
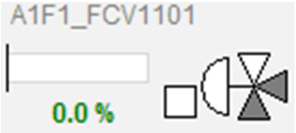
Genie Icons

Refer to Genie Icons, page 63.

Representation

The table describes the genies of the `gpl_valvecontrol` library:

Graphic symbol	Genie name	Description
	*2way_Horiz*	Two-way control valve shown horizontally.
	*2way_Vert_Left*	Two-way control valve shown vertical left.
	2way_Vert_Right	Two-way control valve shown vertical right.
	*3way_Horiz_Downleft*	Horizontal three-way valve (down to left way when closed, down to right way when open).
	*3way_Horiz_Downright*	Horizontal three-way valve (down to left way when closed, down to right way when open).
	*3way_Horiz_Leftdown*	Horizontal three-way valve (left to down way when closed, left to right way when open).
	*3way_Horiz_Leftright*	Horizontal three-way valve (left to right way when closed, left to down way when open).

Graphic symbol	Genie name	Description
	*3way_Horiz_Rightdown*	Horizontal three-way valve (right to down way when closed, right to left way when open).
	*3way_Horiz_Rightleft*	Horizontal three-way valve right to left way when closed, right to down way when open).
	*3way_Vert_Downright*	Vertical three-way valve (down to right way when closed, down to up way when open).
	*3way_Vert_Downup*	Vertical three-way valve (down to up way when closed, down to right way when open).
	3way_Vert_Rightdown	Vertical three-way valve (right to down way when closed, right to up way when open).
	*3way_Vert_Rightup*	Vertical three-way valve (right to up way when closed, right to down way when open).
	*3way_Vert_Updown*	Vertical three-way valve (up to down way when closed, up to right way when open).
	*3way_Vert_Upright*	Vertical three-way valve (up to right way when closed, up to down way when open).

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

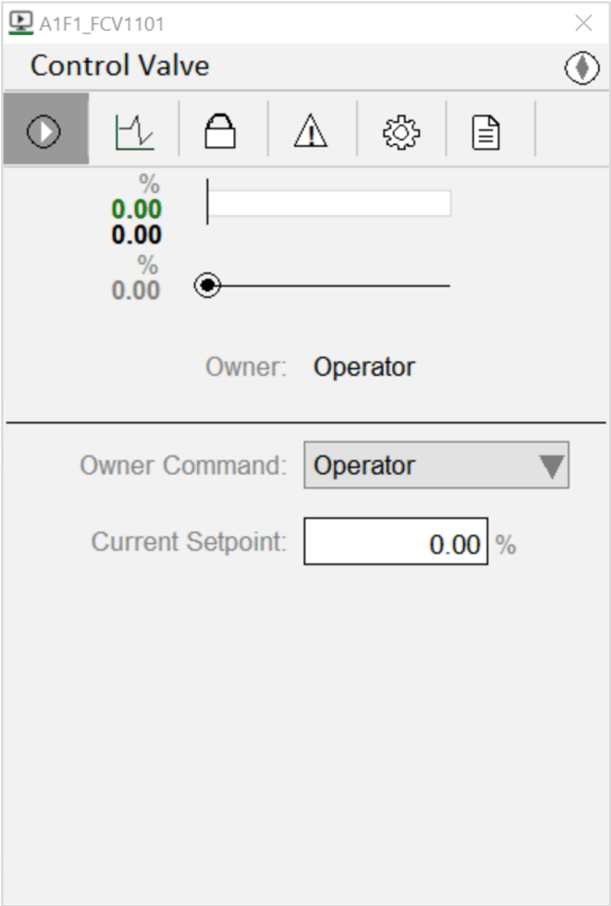
Available Tabs

During operation, clicking a control valve genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
  - Event, page 69
- Tabs for optional functions, which appear only if configured:
  - Interlock, page 87
  - Failure, page 78
  - Measures, page 70

Operator Tab

The figure shows an operator tab:

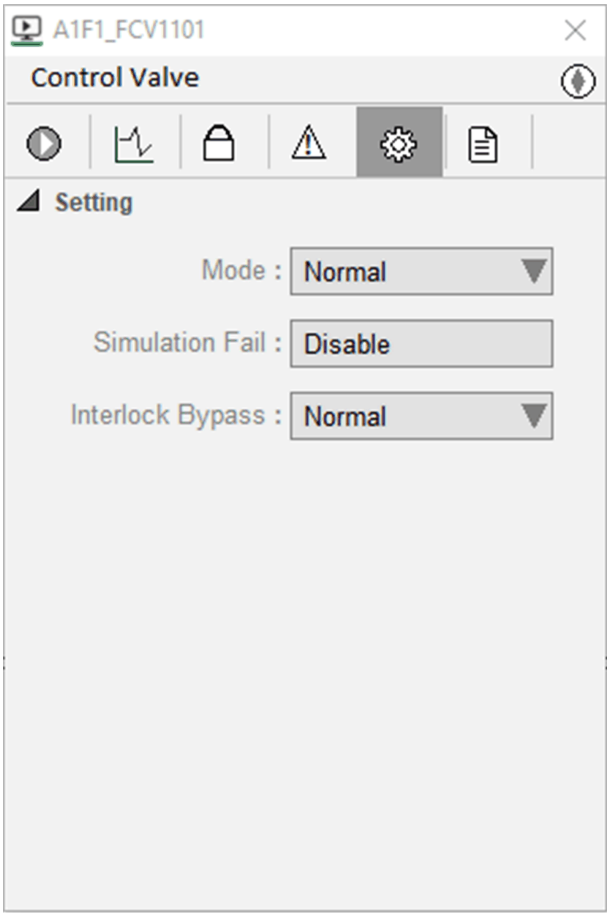


The table describes the control function of an operator tab:

Label	Item Name	Description	Security Parameter, page 354
Owner	Owner	Owner of control valve. Refer to operating owner, page 58.	–
OwnerCom-mand	OwnerCommand	Owner command of control valve (Program, Operator or Local Panel). **NOTE:** Local panel owner is enabled when the virtual panel is selected.	Set owner
Current Setpoint	CurrentSetpoint	Current setpoint of control valve.	–

Engineering Tab

The figure shows an engineering tab:



The table describes the control function of engineering tab:

Label	Item Name	Description	Security Parameter, page 354
Mode	Mode	Allows you to view and change the mode. Refer to operating mode, page 59.	Set mode - override, set mode - simulation, set mode - maintenance, and set mode - out of service **NOTE:** Each configurable mode has the different security parameters
Simulation Fail	SimulationFailActive **NOTE:** The Simulation Fail will only be available when the parameter `Simulation-Fail` is created in the citect parameters. To create, go to **Setup** > **Parameters**. Enter the `GPL.Mode` in **Section Name**, `Simulation-Fail` in **Name** and set the **Value** = 1.	Simulation fail active	set mode - simulation
Interlock Bypass	InterlockBypass	Interlock bypass status Allows you to bypass interlock.	Set bypass conditions, set bypass all interlocks

Items

Overview

This section describes the variables, equipstatus and equipstate variable, abnormal and action required conditions, alarms, events, trends items of `control valve`.

Variables

The table describes the variable items that are used by Supervision components:

Item Name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`InputChannelFailure`	Input channel inoperable	Active/Inactive	*Bit0*
`InterlockActive`	Interlock status	Interlocked/Healthy	*Bit1*
`ProgramSetpointMode`	Setpoint source for program owner	Remote/Sequence	*Bit2*
`OutputChannelFai-lure`	Output channel inoperable	Active/Inactive	*Bit4*
`PositionOpen`	Open status based on analog position	Open/Not Open	*Bit6*
`PositionClosed`	Close status based on analog position	Closed/Not Closed	*Bit7*

Item Name	Description	Enumeration	Address
NotReady	Control valve not ready	–	*Bit8*
FailActive	Control valve inoperable	Active/Inactive	*Bit9*
AlarmActive	Feedback confirmation alarm	Active/Inactive	*Bit10*
FeedbackClosed	Closed feedback	Closed/Not Closed	*Bit11*
FeedbackOpen	Open feedback	Open/Not Open	*Bit12*
CloseLimitEnable	Close limit enable	Enabled/Disabled	*Bit13*
OpenLimitEnable	Open limit enable		*Bit14*
Bit items derived from ConfigurationWord.			
OwnerSelect	Owner of control module	Operator/Program	*Bit0*
InterlockBypass	Interlock bypass command	Bypassed/Normal	*Bit1*
InputSource	Input source selection	Operator/Field	*Bit2*
SimulationFailAc-tive	Feedback of detected failure in simulation mode	–	*Bit12*
Simulation	Simulation mode		*Bit13*
Maintenance	Maintenance mode		*Bit14*
OutOfService	Motor unavailable	Out Of Service/In Service	*Bit15*
IO Items			
StatusWord	Status word	–	CVALVEGP_ST.STW
ConfigurationWord	Configuration word		CVALVEGP_ST.CFGW
Calculated Items			
Equipstatus	Current status of the control valve For more details, refer to EquipStatus calculated variable, page 374.	–	GPL_ ControlValveSta-tus
EquipState	Output information For more details, refer to EquipState calculated variable, page 375.		GPL_ ControlValveE-quipState
Owner	Current owner of the control valve For more details, refer to calculated variable, page 58.		GPL_OwnerLocal
Mode	Current mode of the control valve For more details, refer to calculated variable, page 59.		GPL_Mode
Abnormal	Shows abnormal condition of the control valve For more details, refer to abnormal conditions, page 375.		GPL_Abnormal

Item Name	Description	Enumeration	Address
ActionRequired	Shows action required condition of the control valve For more details, refer to action required conditions, page 375.		GPL_ ActionRequired
Optional Items			
Local panel			
Bit items derived from LPStatusWord.			
LocalPanelMode	Control from field	Enabled/Disabled	*Bit0*
ZeroMode	control valve control disabled		*Bit1*
ControlSystemMode	Control from remote		*Bit2*
LockedOwner	Owner change disabled from faceplate	Locked/Unlocked	*Bit3*
LockedSetpoint	Setpoint change disabled from faceplate		*Bit4*
LocalPanelEnabled	Control from field	Enabled/Disabled	*Bit5*
PushButtonEnabled	Local panel push buttons		*Bit6*
VirtualLocalPane-lEnabled	Local panel enable from control		*Bit7*
LocalPanelSelected	Control from field	Selected/Not Selected	*Bit8*
ZeroSelected	Control disabled		*Bit9*
ControlSystemSelec-ted	Control from remote		*Bit10*
PushButtonOpen	Open push button signal	On/Off	*Bit11*
PushButtonClose	Close push button signal		*Bit12*
Bit items derived from LPConfigurationWord.			
VirtualSwitch	Local panel selection from faceplate	Local/Control System	*Bit0*
IO Items			
LPStatusWord	Local panel status word	–	CVALVELPGP_ST.STW
LPConfigurationWord	Local panel configuration word		CVALVELPGP_ST. CFGW
Failures	Refer to the topic describing the variable items of failures, page 61.		
Interlocks	Refer to the topic describing the variable items of interlocks, page 60.		
Maintenance	Refer to the topic describing the variable items of maintenance, page 59.		

EquipStatus Calculated Variable

The table describes the *EquipStatus* calculated variable:

Equipment Status	Conditions	Enumeration
–	Open x Closed	Unknown
CValve Closed	Not Open x Closed	Closed

Equipment Status	Conditions	Enumeration
CValve Open	Not `Open` x Not `Closed`	Open
CValve Stopped	`Open` x Not `Closed`	Fully Open

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	Conditions	Enumeration
Active with Interlock	`ILCK` x (rOutput > rEngMinScale)	Open Interlocked
Passive with Interlock	`ILCK` x Not (rOutput > rEngMinScale)	Close Interlocked
Active	Not `ILCK` x (rOutput > rEngMinScale)	Open
Passive	Not `ILCK` x Not (rOutput > rEngMinScale)	Close

Abnormal Conditions

The table describes the abnormal conditions of control valve:

Item Name	Description	Address
Bit items derived from Abnormal variable word.		
`AbnormalOwner`	Abnormal owner status appears if the current owner is not a normal owner which configured at the instance level.	*Bit0*
`AbnormalMode`	Abnormal mode status appears if the current operating mode is not normal mode.	*Bit1*
`AbnormalInterlockBy-pass`	Abnormal interlock bypass condition appears if the Interlock bypass is enabled.	*Bit2*
`AbnormalOperatorTab`	Abnormal status of operator tab. This condition appears if the Interlock bypass is enabled.	*Bit8*
`AbnormalInterlockTab`	Abnormal status of interlock tab.	*Bit11*
`AbnormalEngineeringTab`	Abnormal status of engineering tab.	*Bit12*

Action Required Conditions

The table describes the action required conditions of control valve:

Item Name	Description	Address
Bit items derived from ActionRequired variable word.		
`ActionOperatorTab`	Action required status of operator tab.	*Bit8*
`ActionInterlockTab`	Action required status of interlock tab.	*Bit11*

Alarms

The table describes the advanced alarm items that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
Fail	Inoperable control valve	GPL_BitValueCheck (StatusWord, 9)
Alarm	Detection of not confirmed operation	GPL_BitValueCheck (StatusWord, 10)

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item Name	Description
StateAlarm	All changes in the control valve status and output.
ModeAlarm	All changes in the mode and owner.
ActionAlarm	All actions by the operator.

Trends

The table describes the trend items that are used by Supervision components:

Item Name	Expression	Description
PV	CVALVEGP_ST.PV	Present value
SP	CVALVEGP_CFG.LSP	Setpoint
OP	CVALVEGP_ST.LOP	Output

Signals

What's in This Part

Analog Input 378
Analog Output 395
Digital Input 416
Digital Output 428
Range 441
Totalizer 443
Multiple Analog Input 462
Load cell 482
BMXERT1604 516

Overview

This part provides the functionality of the templates, a detailed description of the control (functions, pins, pin layout, and variables of the function blocks), and the Supervision functions of signals.

These components do not reflect any specific installation.

⚠ WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate them.
- Review the implications of communication link interruptions and take actions to mitigate them.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, and applicable codes and regulations.
- Apply local accident prevention and safety regulations and guidelines.[1]
- Test each implementation of a system for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

Analog Input

What's in This Chapter

Template 378
Control 381
Supervision 385

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the analog input.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$AnalogInputGP` template.

Description

General Description

The `$AnalogInputGP` process control module template is designed to read signals, and optionally, signal quality data from analog input modules. Alternatively, it can receive signals from an external variable of type Int or Real.

The template provides linear scaling, cut-off, diagnosis, and simulation functions that you can configure in the **Properties** window of the instance.

During operation, genies and the corresponding faceplate allow you to display the measured value, the set point, ranges, alarms, and so on.

Functional Description

The main features of the `$AnalogInputGP` template are described in the following table:

Function	Description
`Scaling`	The template scales the input signal (usually in raw data) to engineering units through a linear function.
`Cut-Off`	You can configure a minimum value, which is called the cut-off value. If the measured value received from the transmitter is below the cut-off value, the measured value is non-useable. In such case, the cut-off value is being considered.
`Diagnosis`	The template manages the diagnostic status of the signal if the peripherals used provide the signal, and assigns the value that is to be used if the signal is not working properly.
`Mode selection`	The object can be switched to different modes of operation like Maintenance and Override.
`External PV`	The template enables you to connect a signal that is already in engineering units (and therefore does not need scaling, diagnosis, or cut-off operations) while maintaining the simulation function.
`Optimal range`	You can configure the optimal range based on the defined range or Setpoint deviation.

Optional Services

Available optional services are:

- Limit Alarm
- Deviation Alarm
- Rate Alarm

Parameters

Parameters

The `$AnalogInputGP` template provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Range	High range (Raw data)	`Short`	2047	High range of the input signal coming from the channel.
	Low range (Raw data)		0	Low range of the input signal coming from the channel.
	High range	`Float`	100.0	High range of the output measurement (engineering units), corresponding to the maximum value of the channel input (configured for the High Range parameter).
	Low range		0.0	Low range of the output measurement (engineering units), corresponding to the maximum value of the channel input (configured for the Low Range parameter).
	Engineering unit	`String`	%	Engineering units
	Format type		####. #EU	The format type to be used in the supervision.

Analog Input Signals

The table describes the analog input signal parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Include quality signal (applicable for 'Int' data type)	`Boolean`	False	Refer to the topic describing the control DFB, page 381.
	Custom signal name	`String`	–	
	Select analog channel data type **NOTE:** This parameter is not applicable for the `$AnalogInMultiGP` template	`Enum`	Int (0)	

Analog Alarms

For analog alarms, refer to Aalarm, page 108.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Security	Set setpoint	Enum	Operator (1)	Privilege to change setpoint. This parameter is used in Operator Tab, page 389.
	Set mode - override		Supervisor (2)	Privilege to change override mode. This parameter is used in Engineering Tab, page 390.
	Set mode - maintenance		Maintenance (3)	Privilege to set maintenance mode. This parameter is used in Engineering Tab, page 390.
	Set reset		Operator (Confirmed) (10)	Privilege to reset. This parameter is used in Operator Tab, page 389.
	Set parameters		Engineer (5)	Privilege to set parameters. This parameter is used in Operator Tab, page 389.
	Set alarm limits		Operator (1)	Privilege to set alarm limits. This parameter is used in Operator Tab, page 389.
	Set alarm acknowledge		Operator (1)	Privilege to acknowledge alarm. This parameter is used in Alarm Tab, page 67.
Trend	–	–	–	For more details, refer to the topic describing the parameters of trend, page 47.

Asset and Display Parameters

For more details, refer to the topic describing the parameters of asset and display, page 43.

Data

The table describes the data parameters:

Section	Parameter description	Type	Default	Additional remarks
AnalogInput				
Range	Optimal range	Enum	Disable (0)	It disables the optimal range indication.
	Practical range high/low disable	Boolean	True	It disables practical range high and practical low value.
	Practical range disable			It disables the practical range indication.
Alarm	Channel failure description	String	@(Channel Failure Alarm)	A detail description for alarms.
	Channel failure severity	Enum	Low (4000)	It allows to set the criticality (Critical (1000), High (2000), Medium (3000) and Low (4000)) of alarm.
	Channel failure group	String	Channel	Group name for rapid alarm filtering.
Deadband	Present valve	Float	0.0	–
Trend	Disable trend	Boolean	False	–

Section	Parameter description	Type	Default	Additional remarks
Events	Enable log	`Boolean`	False	Enable event logging for asset status.
Historize	Present value	`Boolean`	False	–
	Sensor value			–
	Ranges			–
	Status/ commands			–

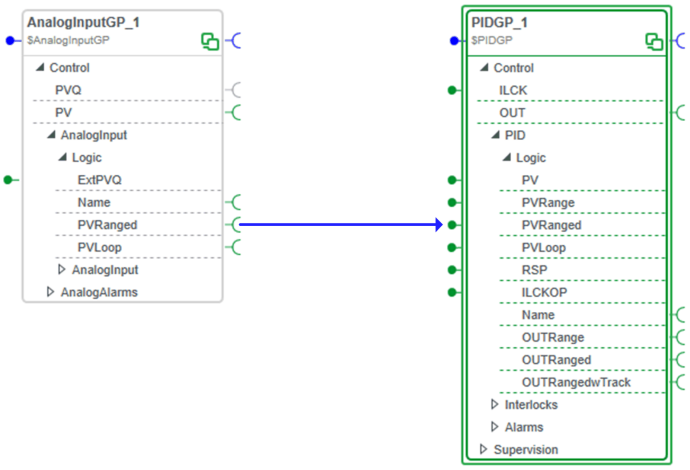
For parameters of AAlarm, refer to, page 108.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `AnalogInputGP` (also applicable for `AnalogInputCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



NOTE: The PVLoop interface has to be linked to the PID if the user wants to navigate from the analog input faceplate to the PID faceplate in supervision.

NOTE: For analog input and PID faceplate navigation, it is recommended to define the parameter `Process Page Name` where these instances will be located in graphic page else during navigation the process display area will not show the content. For parameter description, refer to, page 43.

For interface identifiers, refer to the control module template interfaces, page 49.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

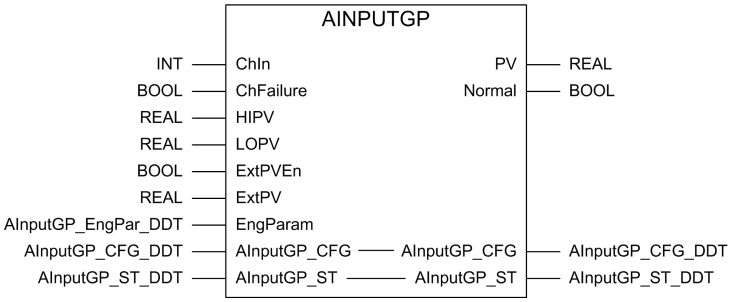
Overview

This chapter describes the `AINPUTGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Name	Type	Description
ChIn	INT	Input signal. Corresponds to a value in raw data coming from the input/output peripherals.
ChFailure	BOOL	1 = Indicates to the DFB that the input channel associated with the ChIn input is operational.
HIPV	REAL	High range of the measurement in engineering units (PV output) corresponds to the maximum value (configured in the HIRaw input) of the ChIn input.
LOPV	REAL	Low range of the measurement in engineering units (PV output) corresponds to the minimum value (configured in the LORaw input) of the ChIn input.
ExtPVEn	BOOL	1 = Enables you to configure the DFB to accept an analog input in engineering units (ExtPV). 0 = Enables you to use the input in raw data (ChIn).
ExtPV	BOOL	Input signal in engineering units (refer to the ExtPVEn input pin).
EngParam	AInputGP_EngPar_DDT	Engineering parameters

AInputGP_EngPar_DDT Type

Name	Type	Description
HIRaw	INT	High range of the ChIn input signal.
LORaw	INT	Low range of the ChIn input signal.

Outputs

Output Parameter Description

Name	Type	Description
PV	REAL	Calculated measurement value usually in engineering units.
Normal	BOOL	1 = Normal mode active.

The table describes how the DFB calculates the measurement value *PV* based on the value of the inputs and the AInputGP_ST input/output:

ExtPVEn	AInputGP_ST. CFGW.Override	ChFailure	ChIn	PV is calculated as:
-	ON	-	-	AInputGP_CFG.SIM
ON	OFF	-	-	ExtPV
OFF	OFF	ON	-	BADPV
OFF	OFF	OFF	<= LORaw	LOPV
OFF	OFF	OFF	>LORaw and =<HIRaw	Refer to the linear scaling formula.
OFF	OFF	OFF	>=HIRaw	HIPV

Linear scaling formula:

$$\left(\frac{HIPV - LOPV}{HIRAW - LORAW} \times CHIN\right) + LOPV - \left(\frac{HIPV - LOPV}{HIRAW - LORAW} \times LORAW\right)$$

If the resulting value of the calculation is less than cut-off (and the signal comes from the ChIn input), the PV output is matched to the LOPV input.

Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
AInputGP_CFG	AInputGP_CFG_DDT	Provides the data necessary to configure the DFB usually from the monitoring subsystem.
AInputGP_ST	AInputGP_ST_DDT	Provides the necessary data for monitoring the state of the input analog signal and configuring its simulation when required.

AInputGP_CFG_DDT Type

Name	Type	Description
Override-Value	REAL	If override is selected, then input value is equal to external PV, else input value is equal to channel value.
CHIPV	REAL	High range of the measurement in engineering units (PV output) corresponding to the maximum value (configured in the HIRaw input) of the ChIn input pin. The maximum value that this variable can assume is limited by the value of the HIPV input pin. If the range entered from the monitoring system is incorrect (AINPUTGP_CFG.CLOPV = AINPUTGP_CFG.CHIPV = 0.0), it is initialized with the value of the HIPV input pin.

Name	Type	Description
CLOPV	REAL	Low range of the measurement in engineering units (PV output) corresponding to the minimum value (configured in the LORaw input) of the ChIn input pin. The minimum value that this variable can assume is limited by the value of the LOPV input pin. If the range entered from the monitoring system is incorrect (AINPUTGP_CFG.CLOPV = AINPUTGP_CFG.CHIPV = 0.0), it is initialized with the value of the LOPV input pin.
BadPV	REAL	Default value in engineering units (PV output signal) when the channel is not operational.
CutOffPV	REAL	Value (in engineering units) resulting from the scaling calculation, and below which the value entered into the LOPV input needs to be considered as the measurement value (PV output).
InputVal-ue	REAL	Input value (in engineering units). If external PV is enable from the control, then input value is equal to external PV. If external PV is disable from the control, then input value is equal to channel value.

AInputGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	ChFailure	*Bit0*	BOOL	Refer to the ChFailure input pin, page 382.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
	Override	*Bit0*	BOOL	Refer to the Override, page 384.
	Maint	*Bit14*	BOOL	Refer to the Maint, page 384.
PV	–	–	REAL	Read-only access Refer to the PV output pin, page 383.

Public Variables

Public Variable Description

Variable	Type	Description
SC	AInputGP_SC_DDT	Provides the frequently needed data to monitor the analog input status from the sequential control.

AInputGP_SC_DDT Type

Name	Type	Description
PV	REAL	Read-only access Refer to the PV output pin, page 383.
BadSt	BOOL	Read-only access Indicates whether the input channel is operational. Reproduces the value of the ChFailure input if the ChIn input signal (ExtPVEn = 0) is used.
Normal	BOOL	Read/write access Refer to the Normal output pin, page 383.

Name	Type	Description
`Override`	`BOOL`	Read/write access Enables to set the device to `Override` mode so that the actual position signals (`PV` input) are ignored. It is simulated that the valve is in the desired position (according to the `LOP` output pin) and is functional.
`Maint`	`BOOL`	Read-only access Refer to the `Maint` inputs/outputs pin, page 384.

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the management of analog input.

Genies

Genie Properties

Refer to Genie Properties, page 62.

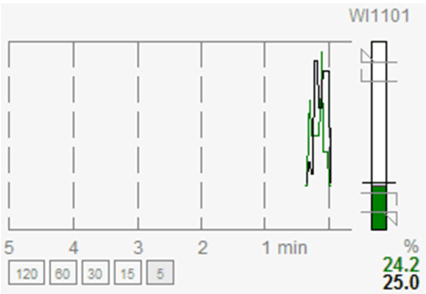
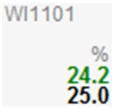
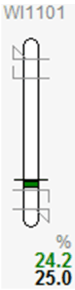
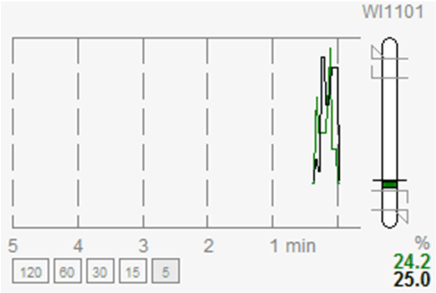
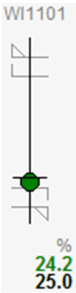
Genie Icons

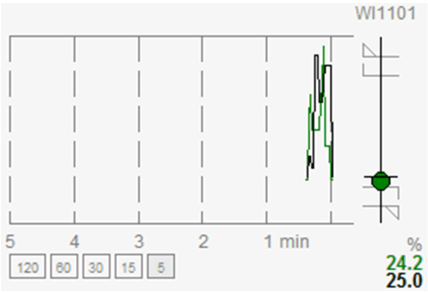
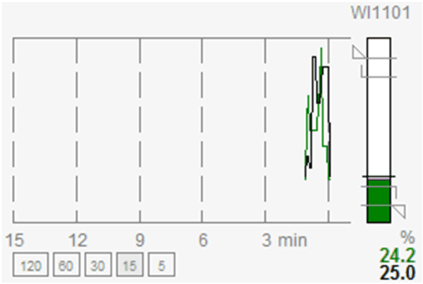
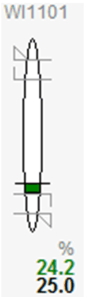
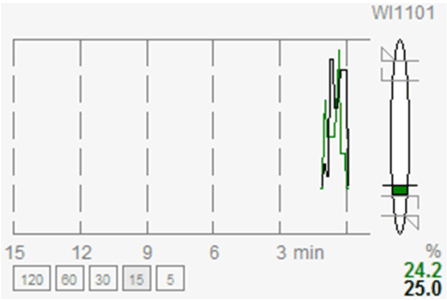
Refer to Genie Icons, page 63.

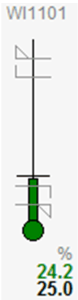
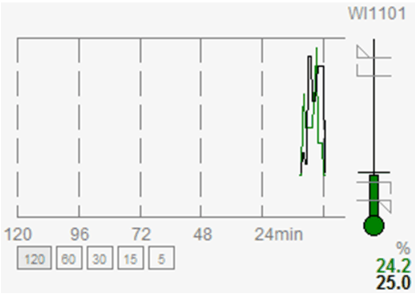
Representation

The table describes the genies of the `gpl_meters` library:

Graphic symbol	Genie name	Description
	Bar_PVSP_Horiz	Displays the label on a horizontal bar with engineering units, present value (PV), and setpoint (SP).
	Bar_PVSP_Vert	Displays the label on a vertical bar with engineering units, present value (PV), and setpoint (SP).

Graphic symbol	Genie name	Description
	*Bar_PVSP_Trend_Vert*	Displays the label on a vertical bar with engineering units, present value (PV), and setpoint (SP). In addition, the symbol displays trends and allows you to select the trend period in minutes.
	Deviation_PVSP	Displays the deviation on a vertical bar with engineering units, present value (PV), and setpoint (SP).
	*Indicator_PVSP*	Displays the label, engineering units, present value (PV), and setpoint (SP).
	*Analyzer_PVSP*	Displays the analyzer on a vertical bar with engineering units, present value (PV), and setpoint (SP).
	*Analyzer_PVSP_Trend*	Displays the analyzer on a vertical bar with engineering units, present value (PV), and setpoint (SP). In addition, the symbol displays trends and allows you to select the trend period in minutes.
	*Flow_PVSP*	Displays the flow on a vertical bar with engineering units, present value (PV), and setpoint (SP).

Graphic symbol	Genie name	Description
	*Flow_PVSP_Trend*	Displays the flow on a vertical bar with engineering units, present value (PV), and setpoint (SP). In addition, the symbol displays trends and allows you to select the trend period in minutes.
	Level_PVSP	Displays the level on a vertical bar with engineering units, present value (PV), and setpoint (SP).
	*Level_PVSP_Trend*	Displays the level on a vertical bar with engineering units, present value (PV), and setpoint (SP). In addition, the symbol displays trends and allows you to select the trend period in minutes.
	*Pressure_PVSP*	Displays the pressure on a vertical bar with engineering units, present value (PV), and setpoint (SP).
	*Pressure_PVSP_Trend*	Displays the pressure on a vertical bar with engineering units, present value (PV), and setpoint (SP). In addition, the symbol displays trends and allows you to select the trend period in minutes.

Graphic symbol	Genie name	Description
	*Temperature_PVSP*	Displays the temperature on a vertical bar with engineering units, present value (PV), and setpoint (SP).
	*Temperature_PVSP_Trend*	Displays the temperature on a vertical bar with engineering units, present value (PV), and setpoint (SP). In addition, the symbol displays trends and allows you to select the trend period in minutes.
NOTE: Setpoint is optional parameter and it is hidden, if setpoint item is not available.		

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

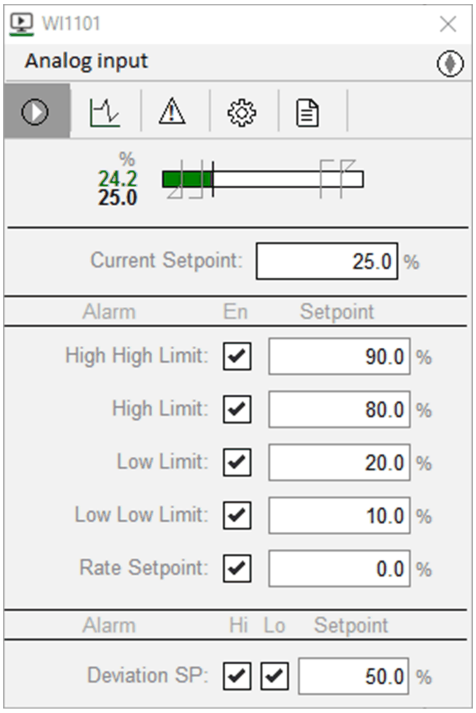
Available Tabs

During operation, clicking an analog input genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
  - Event, page 69
- Tabs for optional functions, which appear only if configured:
  - Measures, page 70

Operator Tab

The figure shows an operator tab:



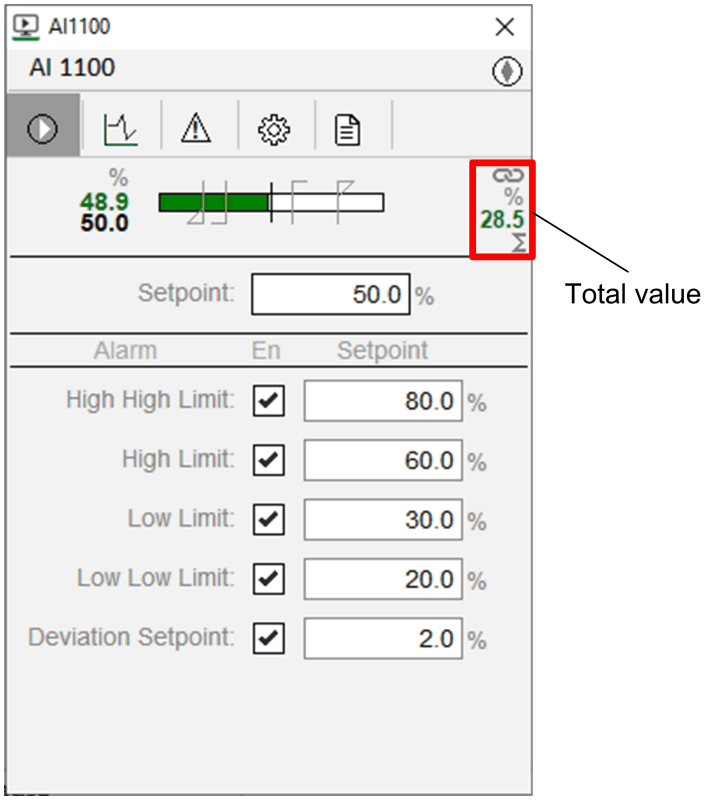
Following actions can be performed from an operator tab:

- **En**: Allows you to enable or disable the evaluation of an alarm conditions at the controller level.
- **Setpoint**: Allows you to set the setpoint.

The table describes the control function of an operator tab:

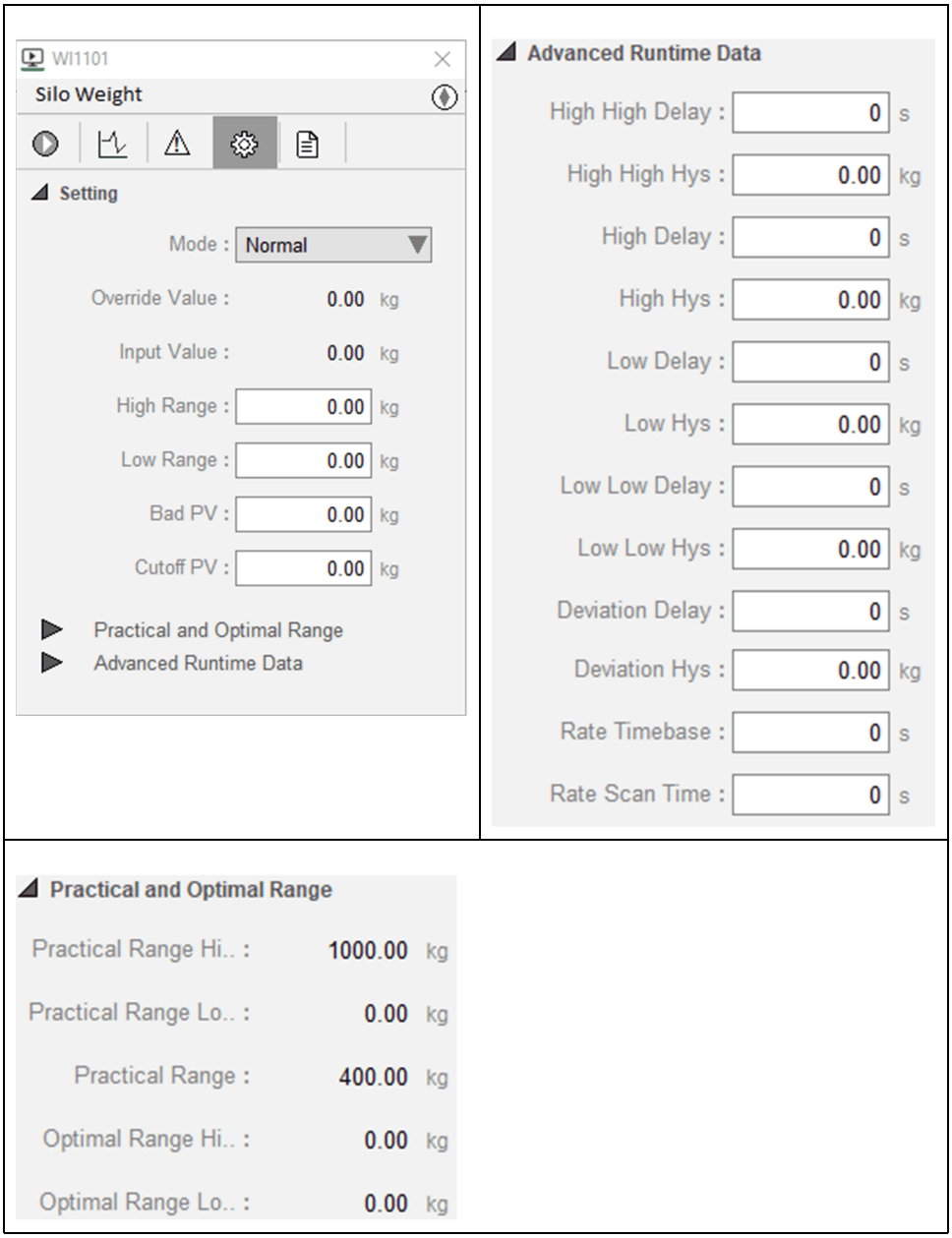
Lable	Item name	Description	Security parameter, page 380
Current Setpoint	SP	Reference setpoint	Set setpoint
High High Limit	HighHighLimit, HighHighEnable	Very-high-level setpoint	
High Limit	HighLimit, HighEnable	High-level setpoint	
Low Limit	LowLimit, LowEnable	Low-level setpoint	
Low Low Limit	LowLowLimit, Add LowLowEnable Item	Very-low-level setpoint	
Rate Setpoint	RateEnbale, RateSetpoint	Rate setpoint	Setmode - override
Deviation SP	SPDeviationLimit, HighDeviationEnable, LowDeviationEnable	Deviation limit setpoint	Set setpoint

If the analog input is linked to totalizer in Process Expert then total value will be displayed on faceplate as shown in below figure:



Engineering Tab

The table shows an engineering tab:



The table describes the control function of an engineering tab:

Legend	Item name	Description	Security parameter, page 380
Setting			
Mode	Mode	Allows you to view and change the mode. Refer to Operating Mode, page 59.	Set mode - override, set mode - maintenance.
Override Value	OverrideValue	If override is selected, then input value is equal to external PV, else input value is equal to channel value.	Set mode - override
Input Value	InputValue	Input value shows the current channel value.	–
High Range	PVHigh	Configurable high range.	Set parameter
Low Range	PVLow	Configurable low range.	Set parameter
Bad PV	PVBad	Value to consider when channel inoperable.	Set parameter
Cutoff PV	PVCutOff	Minimum reliable measure.	Set setpoint
Practical and Optimal Range			

Legend	Item name	Description	Security parameter, page 380
Practical Range High	PRHigh	Practical range high	–
Practical Range Low	PRLow	Practical range low	–
Practical Range	PR	Practical range	Set setpoint
Optimal Range High	ORHigh	Optimal range high	–
Optimal Range Low	ORLow	Optimal range low	–
Advance Runtime Data			
High High Delay	HighHighLimitDelay	Delay high high	Set parameters
High High Hys	HighHighLimitHyste-resis	Hysteresis high high	
High Delay	HighLimitDelay	Delay high	
High Hys	HighLimitHysteresis	Hysteresis high	
Low Delay	LowLimitDelay	Delay low	
Low Hys	LowLimitHysteresis	Hysteresis low	
Low Low Delay	LowLowLimitDelay	Delay low low	
Low Low Hys	LowLowLimitHyste-resis	Hysteresis low low	
Deviation Delay	DeviationDelay	Delay for enabling deviation.	
Deviation Hys	DeviationHysteresis	Hysteresis value for deviation alarm.	
Rate Timebase Delay	RateTimebase	Time base rate setpoint	
Rate Scan Time	RateScanTime	Time scan rate setpoint	

Items

Overview

This section describes the variables, abnormal and action required conditions, alarms, events, and trends items of analog input.

Variables

The table describes the variable items that are used by supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`ChannelFailure`	Input channel inoperable	Healthy/ Unhealthy	*Bit0*
Bit items derived from ConfigurationWord.			
`InputSource`	Input source selection	Field/Operator	*Bit0*
`Maintenance`	Maintenance mode	–	*Bit14*
IO Items			
`StatusWord`	Status word	–	`AINPUTGP_ST.STW`
`ConfigurationWord`	Configuration word	–	`AINPUTGP_ST.CFGW`

Item name	Description	Enumeration	Address
Calculated Items			
`Mode`	Current mode of the analog input For more details, refer to calculated variable, page 59.	–	`GPL_ModeSignalInput`
`Abnormal`	Shows abnormal condition of the analog input For more details, refer to abnormal Conditions, page 393.	–	`GPL_SignalAbnormal`
`ActionRequired`	Shows action required condition of the motor For more details, refer to action required conditions, page 393.		`GPL_ActionRequired`

Abnormal Conditions

The table describes the abnormal conditions of analog input:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
`AbnormalMode`	Abnormal mode status appears, if the Mode is not normal.	*Bit1*
`AbnormalHighHighEnable`	This condition appears, if high-high alarm is not enabled.	*Bit2*
`AbnormalHighEnable`	This condition appears, if high alarm is not enabled.	*Bit3*
`AbnormalHighDeviatio-nEnable`	This condition appears, if deviation alarm is not enabled.	*Bit4*
`AbnormalLowLowEnable`	This condition appears, if low-low alarm is not enabled.	*Bit5*
`AbnormalLowEnable`	This condition appears, if low alarm is not enabled.	*Bit6*
`AbnormalLowDeviatio-nEnable`	This condition appears, if deviation alarm is not enabled.	*Bit7*
`AbnormalOperatorTab`	Abnormal status of operator tab. This condition appears, if any of alarm is not enabled.	*Bit8*
`AbnormalEngineeringTab`	Abnormal status of engineering tab. This condition appears if the Mode is not normal.	*Bit12*

Action Required Conditions

The table describes the action required conditions of motor:

Item name	Description	Address
Bit items derived from ActionRequired variable word.		
`ActionOperatorTab`	Action required status of operator tab.	*Bit8*
`ActionInterlockTab`	Action required status of interlock tab.	*Bit11*

Alarms

The table describes the alarms that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
ChannelFailure	Channel failure alarm	GPL_BitValueCheck (StatusWord, 1)

Optional Services:

- Refer to AAlarm, page 114

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
StateAlarm	All changes in the anaolog input status and output.
ModeAlarm	All changes in the mode and owner.
ActionAlarm	All actions by the operator.

Trends

The table describes the trend items that are used by Supervision components:

Item name	Expression	Description
PV	AINPUT1GP_ST.PV	Present value
SP	DALARMGP_CFG.SP	Setpoint
HighLimit	LALARMGP_CFG.SPH	High level setpoint
HighHighLi-mit	LALARMGP_CFG.SPHH	Very high level setpoint
LowLimit	LALARMGP_CFG.SPL	low level setpoint
LowLowLimit	LALARMGP_CFG.SPLL	Very low level setpoint
ORHigh	Based on selection, refer to ORHigh, page 392.	Optimal high range
ORLow	Based on selection, refer to ORLow, page 392.	Optimal low range

Analog Output

What's in This Chapter

Template 395
Control 399
Optional Service - Analog Output Local Panel (AOUTPUTLPGP)..... 403
Supervision..... 407

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the analog output.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$AnalogOutputGP` template.

Description

General Description

The `$AnalogOutputGP` process control module template is used to condition an analog signal usually associated with a physical output.

The process template provides scaling, interlocking, diagnosis, and owner management functions that you can configure in the **Properties** window of the instance.

Function Description

The main functions of the analog output template are described in the following table:

Function	Description
Scaling	The template allows you to scale the input signal in engineering units (for example, %) to raw data through a linear function.
Diagnosis	The template manages the diagnostic status of the signal if the periphery used provides the signal.
Interlocking	The template enables to move to the defined position if an active interlock that requires this move is detected. An interlocking bypass function is available.
Owner	The template allows you to configure owner (Operator or Program). As a result, it is responsible for setting the set-point to the desired position.
Mode selection	The object switched to different modes of operation like Out Of Service, Simulation and Maintenance.

Optional Services

Available optional services are:

- **Interlocks**

- **Local Panel**

Parameters

Parameters

The `$AnalogOutputGP` template provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default value	Additional remarks
Range	**High range (Raw)**	`Short`	*2047*	High range of the input signal coming from the channel.
	Low range (Raw)		*0*	Low range of the input signal coming from the channel.
Set-point	**Force remote setpoint**	`Boo-lean`	*FALSE*	–

Analog Output Signals

The table describes the analog out signal parameters:

Section	Parameter description	Type	Default value	Additional remarks
Configura-tion	**Included signal quality**	`Boolean`	*FALSE*	Refer to the topic describing the control DFB, page 399.
	Custom signal name	`String`	–	

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Local Panel (Logic)

For more details, refer to the topic describing the parameters of logic, page 45.

Local Panel (Digital Input)

The table describes the digital output:

Name	Description
`LocalMode`	Local panel selection signal from mode selector switch.
`ZeroMode`	Zero/Off selection signal from mode selector switch.
`CSMode`	Control selection signal from selector switch.
`OpenPB`	Start signal from local panel.
`ClosePB`	Stop signal from local panel.

For more details, refer to the topic describing the parameters of digital inputs, page 44.

Local Panel (Digital output)

The table describes the digital output:

Name	Description
PBLight	Output signal to indicate that push buttons are operational.

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default value	Additional remarks
Operation	**Normal owner**	Enum	*Program and casecade (5)*	Privilege to change setpoint. This parameter is used in Operator Tab, page 410.
Security	**Set owner**	Enum	*Operator (1)*	Privilege to change owner. This parameter is used in Operator Tab, page 410.
	Set setpoint		*Operator (1)*	Privilege to change setpoint. This parameter is used in Operator Tab, page 410.
	Set mode - Simulation		*Engineer (5)*	Privilege to change simulation mode. This parameter is used in Engineering Tab, page 411.
	Set mode - Maintenance		*Maintenance (3)*	Privilege to set maintenance mode. This parameter is used in Engineering Tab, page 411.
	Set mode - Out of service		*Supervisor (2)*	Privilege to change out of service mode. This parameter is used in Engineering Tab, page 411.
	Set reset		*Operator (Confirmed) (10)*	Privilege to reset. This parameter is used in Operator Tab, page 410.
	Set parameters		*Engineer (5)*	Privilege to change parameters. This parameter is used in Operator Tab, page 410.
	Set alarm limits		*Operator (1)*	Privilege to set alarm limits. This parameter is used in Operator Tab, page 410.
	Set alarm acknowledge		*Operator (1)*	Privilege to acknowledge alarm. This parameter is used in Alarm Tab, page 67.
	Set bypass conditions		*Supervisor (2)*	Privilege to set bypass conditions mode. This parameter is used in Engineering Tab, page 411.
	Set bypass all interlocks		*Supervisor (2)*	Privilege to bypass all interlocks. This parameter is used in Engineering Tab, page 411.
Trend	**–**	–	–	For more details, refer to the topic describing the parameters of trend, page 47.

Data

The table describes the data parameters:

Section	Parameter description	Type	Default value	Additional remarks
Alarm	**Channel failure description**	`String`	*@(Channel Failure Alarm)*	A detail description for alarms.
	Channel failure group	`String`	*Channel*	Group name for rapid alarm filtering.
	Channel failure severity	`Enum`	*Low (4000)*	It allows to set the criticality of alarm.
Trend	**Disable setpoint**	`Boolean`	*FALSE*	–
	Disable output			–
Events	**Enable log**	`Boolean`	*FALSE*	Enable event logging for asset status.
Historize	**Setpoint/output**	`Boolean`	*FALSE*	–

Asset and Display Parameters

For more details, refer to the topic describing the parameters of asset and display, page 43.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Local Panel (Logic)

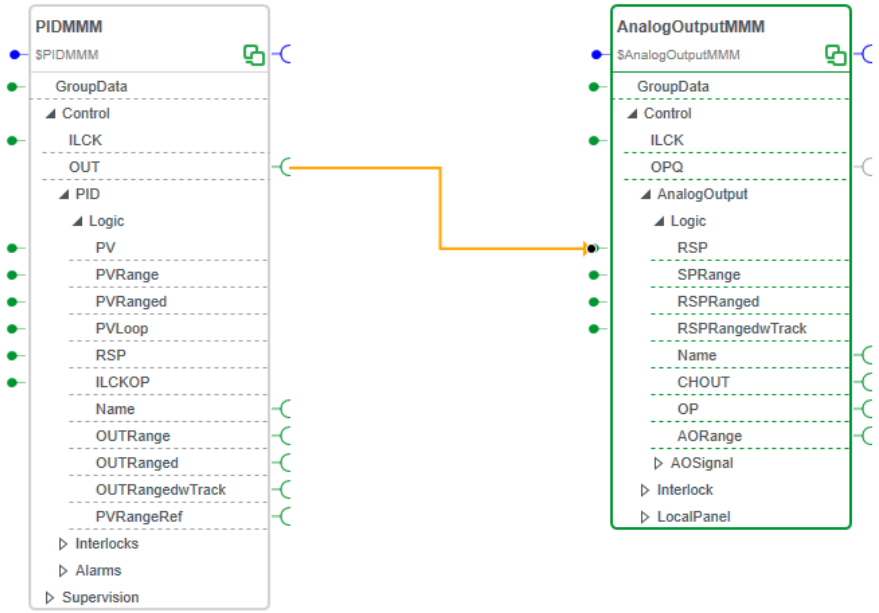
For more details, refer to the topic describing the parameters of logic, page 45.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

The following figure depicts the `$AnalogOutputGP` (also applicable for `$AnalogOutputCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



The following table lists the link interface that creates link symbol on faceplate:

Interface name	Runtime parameter
RSP	SetpointNavigate
RSPRanged	
RSPRangedwTrack	
OP	OPNavigate

For more details, refer to Supervision chapter > Faceplate > Navigation Link of General Purpose Library User Guide.

Refer to common control module template interface, page 47 for more details on common interfaces.

The table describes the control module template interfaces:

Interface identifier	Type/Role	Description	Example
OP	*$Real/Def*	Analog output in engineering units interface to other template.	Analog output OP interface connected to real variable interface

Control

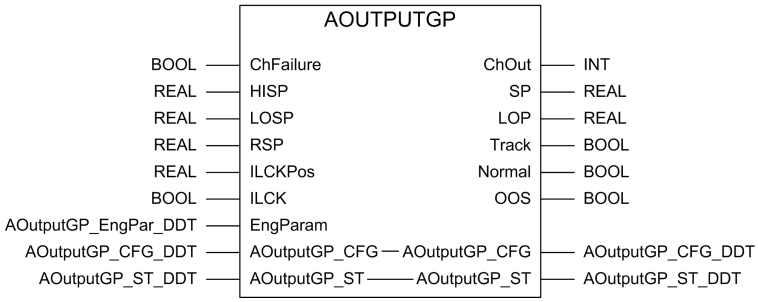
Overview

This chapter describes the AOUTPUTGP DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the control.



Inputs

Input Parameter Description

Parameter	Type	Description
ChFailure	BOOL	Indicates to the DFB whether the output channel associated with the ChOut output is operational (1) or not (0).
HISP	REAL	High range of the set-point (SP) in engineering units corresponds to the maximum value (configured in the HIRaw input) of the ChOut output.

Parameter	Type	Description
LOSP	REAL	Low range of the set-point (SP) in engineering units corresponds to the minimum value (configured in the LORaw input) of the ChOut output.
RSP	REAL	Remote set-point. Usually set by the continuous control; for example, by a PID controller output.
ILCKPos	REAL	Position to be considered during interlock.
ILCK	BOOL	1 = Interlocks the device at the defined position.
EngParam	AOut-putGP_ EngPar_ DDT	Engineering parameters

AOutputGP_EngPar_DDT Type

Name	Type	Description
HIRaw	INT	High range of the ChOut output signal.
LORaw	INT	Low range of the ChOut output signal.

Outputs

Output Parameter Description

Parameter	Type	Description		
ChOut	INT	Calculated value of the output in raw data. Output calculation based on the LOP value is detailed in the following table:		
		LOP	**ChOut**	
		<= LOSP	LORaw	
		> LOSP and < HISP	Is calculated with the following linear formula.	
		>= HISP	HIRaw	
		NOTE: In Simulation mode ChOut holds the last value.		
SP	REAL	Current set-point. Set-point calculation based on the value of the input value and the AOutputGP_ST input/output is detailed in the following table:		
		Owner (OFF: Program, ON: Operator)	**Rem**	**SP**
		OFF	OFF	SC.LSP
		OFF	ON	RSP
		ON	-	AOutputGP_CFG.LSP
LOP	REAL	Output in engineering units. Setpoint calculation based on the SC.ILCKD public variable value is detailed in the following table:		
		SC.ILCKD	**OP**	
		ON	ILCKPos	
		OFF	SP	
Track	BOOL	Determines whether the remote set-point (RSP input) is one being considered (0) or not (1) for the ChOut output calculation. 1 = Report whether the control loop is open to the algorithm connected to the RSP input (for example, a PID) so that it can readjust itself based on the output that is really taking place (the one present at the LOP output).		

Parameter	Type	Description				
		ILCKD	Owner	Rem	OOS	Track is calculated as:
		-	-	-	ON	OFF
		ON	-	-	OFF	ON
		OFF	OFF	OFF	OFF	ON
		OFF	OFF	OFF	OFF	OFF
		OFF	ON	-	OFF	ON
Normal	BOOL	Indicates the equipment operating is in normal mode.				
OOS	BOOL	1 = Indicates the equipment is out of service.				

Linear scaling formula:

$$\left(\frac{HIRAW - LORAW}{HISP - LOSP} \times SP\right) + LORAW - \left(\frac{HIRAW - LORAW}{HISP - LOSP} \times LOSP\right)$$

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
AOutputGP_CFG	AOutputGP_CFG_DDT	Provides the necessary data to control the DFB.
AOutputGP_ST	AOutputGP_ST_DDT	Provides the data necessary to monitor the DFB status.

AOutputGP_CFG_DDT Type

Name	Type	Description
LSP	REAL	Read/write access This variable indicates setpoint and is calculated internally in the DFB as long as the owner is not the Operator (is set to match the SP output). In which case, you can modify it from the monitoring subsystem.

AOutputGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	ChFailure	*Bit0*	BOOL	Refer to the ChFailure input pin, page 399.
	ILCKD	*Bit1*	BOOL	Refer to ILCKD, page 402.
	Rem	*Bit2*	BOOL	Refer to Rem, page 402.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
	Owner	*Bit0*	BOOL	Read/write access Enables to configure whether the set-point is set by the:

Name	Bit Name	Bit	Type	Description
				0 Program 1 Operator
	ILCKBP	*Bit1*	BOOL	Read/write access Allows the interlock to be bypassed (1).
	Sim	*Bit13*	BOOL	Refer to Sim, page 402.
	Maint	*Bit14*	BOOL	Refer to Maint, page 402.
	OOS	*Bit15*	BOOL	Refer to the OOS output pin, page 400.
LOP	–	–	REAL	Refer to the LOP output pin, page 400.

Public Variables

Public Variable Description

Variable	Type	Description
SC	AOutputGP_SC_DDT	Provides the frequently needed data to monitor and control the analog output status from the sequential control.

AOutputGP_SC_DDT Type

Name	Type	Description
LSP	REAL	Read/write access Allows the assignment of the local set-point for the sequential control if the owner is the Program (Owner input/output is set to 0) and the selected set-point is the Local on (SC.Rem public variable is set to 0). Otherwise, the current set-point (SP output) is continuously copied to this variable.
SP	REAL	Read-only access Refer to the SP output pin, page 400.
LOP	REAL	Read-only access Logical output.
Rem	BOOL	Read/write access Allows the DFB to be configured to remote setpoint (RSP = 1) or local setpoint (LSP = 0).
Track	BOOL	Read-only access Refer to the Track output pin, page 400.
BadSt	BOOL	Read-only access Indicates whether the input channel is operational or not. Reproduces the value of the ChFailure input value.
ILCKD	BOOL	Read-only access The signal evaluation depending on the ILCK input and the AOutputGP_ST.CFGW.ILCKBP input/output is shown in the following table:

ILCK	ILCKBP	SC.ILCKD is calculated as:
OFF	-	OFF
ON	OFF	ON
ON	ON	OFF

Name	Type	Description
Owner	BOOL	Read/write access Refer to the Owner inputs/outputs pin, page 401.
Normal	BOOL	Read/write access Refer to the Normal output pin, page 400.
Sim	BOOL	Read/write access Enables to set the valve to simulation mode so that the actual position signals (PV input) are ignored. It is simulated that the equipment is in the desired position (according to the LOP output pin) and the ChOut output pin holds the last value.
Maint	BOOL	Read-only access 1 = Maintenance mode active. Allows the operation of the equipment for maintaining and testing. Interlock conditions are bypassed.
OOS	BOOL	Read/write access Refer to the OOS inputs/outputs pin, page 401.

Optional Service - Analog Output Local Panel (AOUTPUTLPGP)

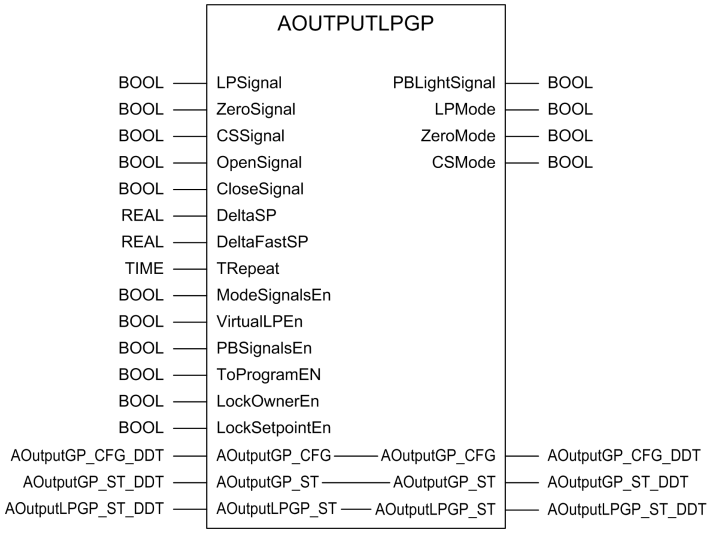
Overview

This chapter describes the optional services - local panel (AOUTPUTLPGP) DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
LPSignal	BOOL	1 = The local panel mode switch is in the Local Panel position. Refer to the table included with the description of the LPMode output pin, page 405.
ZeroSignal	BOOL	1 = The local panel mode switch is in the zero position. This signal is optional even if a mode switch is available on the local panel. When this signal is a logical high, user is not allowed to operate the device from the faceplate and also from the DFB. Hence, all the functions in the faceplate will be disabled. Refer to the table included with the description of the LPMode output pin, page 405.
CSSignal	BOOL	1 = The local panel mode switch is in the Control System position. Refer to the table included with the description of the LPMode output pin, page 405.
OpenSignal	BOOL	1= Indicates to the DFB that the OPEN push button on the local panel is pressed.
CloseSignal	BOOL	1 = Indicates to the DFB that the CLOSE push button on the local panel is pressed.
DeltaSP	REAL	Absolute set-point increase or decrease value when the OPEN or CLOSE push button is pressed.
DeltaFastSP	REAL	Absolute set-point increase or decrease value when the OPEN or CLOSE push buttons are being held down.
TRepeat	TIME	Indicates the maximum time that should elapse before the system interprets that a command has been repeated when the push button is being held down. This time also determines when a DeltaFastSP increase or decrease has to be applied when the OPEN or CLOSE push button is being held down.
ModeSignalsEn	BOOL	1 = Enables the use of the Local Panel/Zero/Control System (or Local Panel - Control System) mode switch on the local panel. Usually configured in development phase based on the characteristics of the local panel being used. If the mode switch is enabled, the LPSignal, ZeroSignal, and CSSignal inputs are considered to determine the operating modes of the local panel. Refer to the table included with the description of the LPMode output pin, page 405. **NOTE:** *ModeSignalsEn* input pin has higher priority than the *VirtualLPEn* input pin signal.
VirtualLPEn	BOOL	1 = This signal enables the operator to select the Local/Control System mode of operation from the HMI, so that the push button signals are enabled for operation. **NOTE:** When the owner is program, the Local/Control System mode selection drop-down list will be visible but disabled for operation however, when the owner is operator the Local Panel drop-down list will be accessible for operation. Refer to the table included with the description of the LPMode output pin, page 405.
PBSignalsEn	BOOL	1 = Enables to accept OPEN and CLOSE push button signals from the local panel. This signal is not applicable if the mode switch is enabled (refer to the ModeSignalsEn input pin). This means that the push button signals are considered in the Local Panel mode. When the mode switch is disabled (ModeSignalsEn = 0), PBSignalsEn signal enables/disables the push buttons from control system itself based on relevant process conditions. SC.DisableLP public variable is also considered for determining whether the push buttons are enabled or not when *LPMode* pin signal is high, as shown in the following table.

VirtualLPEn	ModeSignalsEn	PBSignalsEn	SC.DisableLP	Push buttons enabled?
OFF	OFF	OFF	-	NO
OFF	OFF	-	ON	NO
OFF	OFF	ON	OFF	YES
OFF	ON	-	-	YES
ON	OFF	OFF	-	YES

Therefore, you can use this signal to enable/disable the push buttons on a simple local panel that does not feature a mode switch (Local Panel/Zero/Control System) based on the relevant process conditions.

ToProgramEn	BOOL	1 = Change to program mode when control is switched to control system mode and VirtualLPEn = 0.
LockOwnerEn	BOOL	1 = Disables the access of the owner of analog output remaining as Operator while the local panel mode is Local Panel. The following table shows how the owner of analog output is evaluated based on ToProgramEn and LockOwnerEn inputs and the local panel operating mode:

Local Panel Mode	ToProgramEn	LockOwnerEn	Owner
Zero	-	-	Operator

Parameter	Type	Description			
		Switch to Local Panel	-	-	Operator
		Local Panel	-	ON	Operator
		Switch to Control System	ON	-	Program
		In remaining cases, owner of the analog output ceases to be defined from the `AOUTPUTLPGP` DFB and it becomes possible to change it from monitoring (HMI) system with the command word in the `AOUTPUT_ST.CFGW` input/output of the corresponding `AOUTPUTGP` DFB. In any case, consider that the analog output owner that had been set before switching to the local panel mode is not memorized.			
`LockSetPointEn`	`BOOL`	1 = Disables the setpoint of the analog output and lock it into the one defined on the local panel while the local panel mode is local panel and the device owner is the Operator. User can switch to Program mode based on the signals that have been previously described even if the local panel mode is set to Local Panel. Therefore, user can activate (1) this signal to lock setpoint operation from the monitoring system while the analog output is controlled from the local panel.			

Outputs

Output Parameter Description

Parameter	Type	Description						
`PBLightSig-nal`	`BOOL`	1 = The push buttons on the local panel are fully operational, that is, the local panel is in Local Panel mode (either because the mode is selected with the corresponding switch or because the push buttons are enabled in the event that there is no selector switch available or the *VirtualLPEn* is high with the local panel mode enabled).						
`LPMode`	`BOOL`	1 = The local panel mode is in Local Panel. The following table shows how the local panel operating mode is determined based on the `ModeSignalsEn`, `VirtualLPEn`, `PBSignalsEn`, `LPSignal`, `CSSignal` and `ZeroSignal`, input signals:						
		`Mode-Signal-sEn`	**`Vir-tualLPEn`**	**`PBSigna-lsEn`**	**`LPSignal`**	**`CSSignal`**	**`ZeroSignal`**	**OUTPUT**
		0	0	0	0	0	0	CS mode
		1	0	0	0	0	0	Zero mode
		1	0	0	1	0	0	LP mode and PBLightSignal
		1	0	0	0	1	0	CS mode
		0	1	1	NA	NA	NA	CS mode
		0	1	0	NA	NA	NA	CS mode
		0	0	1	0	0	0	LP mode and PBLightSignal
		1	1	0	1	0	0	LP mode and PBLightSignal
		1	1	0	0	0	0	Zero mode
		1	1	0	0	0	1	Zero mode
		1	1	0	0	1	0	CS mode
		1	0	1	0	1	0	CS mode
		1	1	1	0	1	0	CS mode
		1	0	0	0	0	1	Zero mode
`ZeroMode`	`BOOL`	1 = The local panel mode is in Zero. User is not allowed to operate the device from the faceplate and also from DFB. Hence, all the functions in the faceplate will be disabled. Refer to the table included with the description of the `LPMode` output.						
`CSMode`	`BOOL`	1 = The local panel mode is in Control System. Refer to the table included with the description of the `LPMode` output.						

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
AOutputGP_CFG	AOutputGP_CFG_DDT, page 401	Data structure belonging to the analog output is to be controlled from the local panel (for detailed information regarding this structure, refer to the AOUTPUTGP DFB, page 399). The structure provides information needed for the operation of the AOUTPUTLPGP DFB. It enables this DFB to control the setpoint for the analog output.
AOutputGP_ST	AOutputGP_ST_DDT, page 401	Data structure corresponding to the analog output is to be controlled from the local panel (for detailed information regarding this structure, refer to the AOUTPUTGP DFB, page 399. The structure provides information needed for the operation of AOUTPUTLPGP DFB and enables this DFB to control the owner of the analog output.
AOut-putLPGP_ST	AOutputLPGP_ST_DDT	Data structure that is used as an interface with the monitoring (HMI) system.

AOutputLPGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	LPMode	*Bit0*	BOOL	Refer to the LPMode output pin, page 405.
	ZeroMode	*Bit1*	BOOL	Refer to the ZeroMode output pin, page 405.
	CSMode	*Bit2*	BOOL	Refer to the CSMode output pin, page 405.
	OwnerLocked	*Bit3*	BOOL	Indicates whether the owner change buttons in the monitoring (HMI) system needs to be locked (1) or not (0). Is activated when: • The Local Panel mode is set to Zero or • It is set to Local Panel and the LockOwnerEn input signal has been configured as active (1).
	SPLocked	*Bit4*	BOOL	Indicates whether the set-point change and START buttons in the monitoring (HMI) system needs to be locked (1) or not (0). Is activated when: • The Local Panel mode is set to Zero or • It is set to Local Panel and the LockSetpointEn input signal has been configured as active (1).
	ModeSignalsEn	*Bit5*	BOOL	Refer to the ModeSignalsEn input pin, page 404.
	PBEnabled	*Bit6*	BOOL	Refer to the PBEnabled input pin, page 404.
	VirtualLPEn	*Bit7*	BOOL	Refer to the VirtualLPEn input pin, page 404.
	LPSignal	*Bit8*	BOOL	Refer to the LPSignalinput pin, page 404.
	ZeroSignal	*Bit9*	BOOL	Refer to the ZeroSignal input pin, page 404.
	CSSignal	*Bit10*	BOOL	Refer to the CSSignal input pin, page 404.
	OpenSignal	*Bit11*	BOOL	Refer to the OpenSignal input pin, page 404.
	CloseSignal	*Bit12*	BOOL	Refer to the CloseSignal input pin, page 404.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read-only access to the data contained in this word.

Name	Bit Name	Bit	Type	Description
	LocalModeOn	*Bit0*	BOOL	Read/write access • 1 = Enables the Local Panel mode. • 0 = Enables the Control System mode.

Public Variables

Public Variable Description

Variable	Type	Description
SC	AOutputLPGP_SC_DDT	Provides the frequently needed data to monitor and control the DFB.

AOutputLPGP_SC_DDT Type

Name	Type	Description
LPMode	BOOL	Read-only access Refer to the LPMode output pin, page 405.
ZeroMode	BOOL	Read-only access Refer to the ZeroMode output pin, page 405.
CSMode	BOOL	Read-only access Refer to the CSMode output pin, page 405.
DisableLP	BOOL	Read/write access 1 = Disables the push buttons on the local panel. Only applicable when a local panel without mode switch configuration is used (ModeSignalsEn = 0).

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the management of analog output.

Genies

Genie Properties

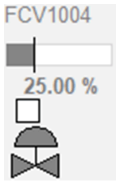
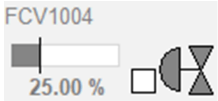
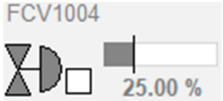
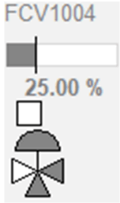
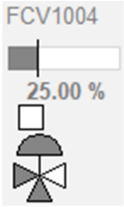
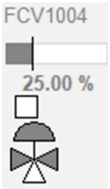
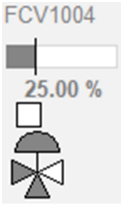
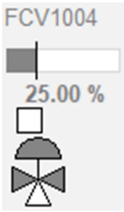
Refer to Genie Properties, page 62.

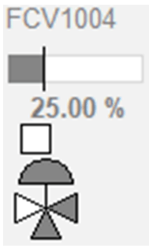
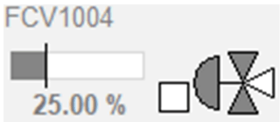
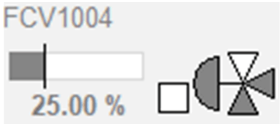
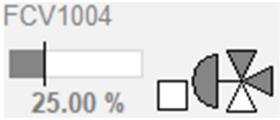
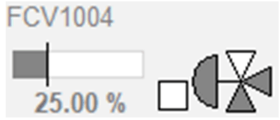
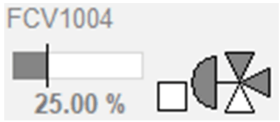
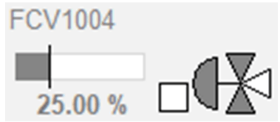
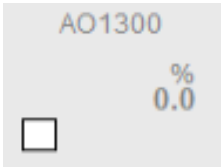
Genie Icons

Refer to Genie Icons, page 63.

Representation

The table describes the genies of the `gpl_analogoutput` library:

Graphic symbol	Genie name	Description
	*2way_horiz*	Two-way control valve shown horizontally.
	*2way_vert_left*	Two-way control valve shown vertical left.
	*2way_vert_right*	Two-way control valve shown vertical right.
	*3way_horiz_downleft*	Horizontal three-way valve (down to left way when closed, down to right way when open).
	*3way_horiz_downright*	Horizontal three-way valve (down to left way when closed, down to right way when open).
	*3way_horiz_leftdown*	Horizontal three-way valve (left to down way when closed, left to right way when open).
	*3way_horiz_leftright*	Horizontal three-way valve (left to right way when closed, left to down way when open).
	*3way_horiz_rightdown*	Horizontal three-way valve (right to down way when closed, right to left way when open).

Graphic symbol	Genie name	Description
	*3way_horiz_rightleft*	Horizontal three-way valve right to left way when closed, right to down way when open).
	*3way_vert_downright*	Vertical three-way valve (down to right way when closed, down to up way when open).
	*3way_vert_downup*	Vertical three-way valve (down to up way when closed, down to right way when open).
	*3way_vert_rightdown*	Vertical three-way valve (right to down way when closed, right to up way when open).
	*3way_vert_rightup*	Vertical three-way valve (right to up way when closed, right to down way when open).
	*3way_vert_updown*	Vertical three-way valve (up to down way when closed, up to right way when open).
	*3way_vert_upright*	Vertical three-way valve (up to right way when closed, up to down way when open).
	*Indicator_op*	Output indicator

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

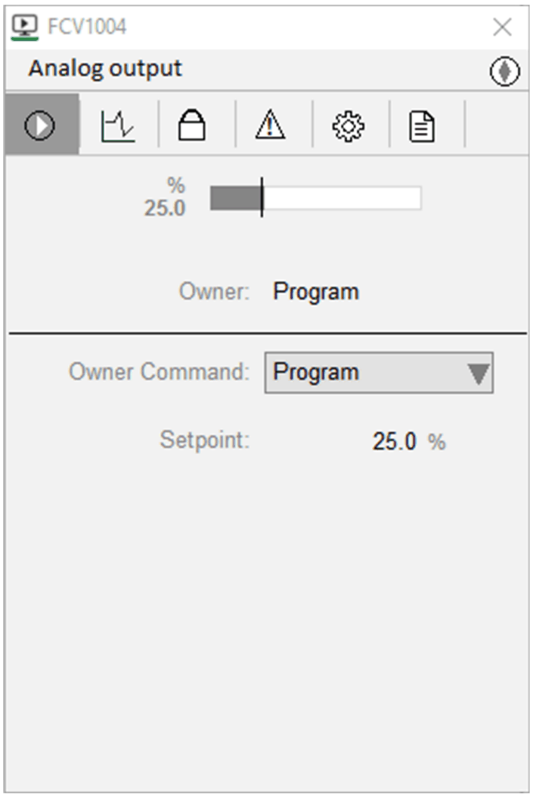
Available Tabs

During operation, clicking an analog output genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
  - Event, page 69
- Tabs for optional functions, which appear only if configured:
  - Measures, page 70

Operator Tab

The figure shows an operator tab:

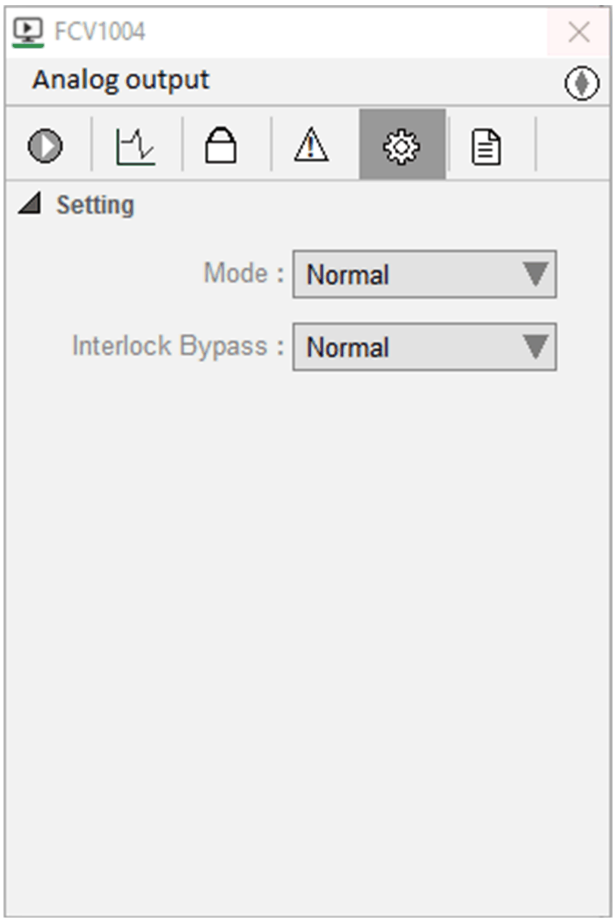


The table describes the control function of an operator tab:

Lable	Item Name	Description	Security parameter, page 397
Owner	Owner	Current owner of analog output. Refer to Owner, page 58.	–
OwnerCommand	OwnerCommand	Owner command of analog output (Program, Operator or Local Panel). **NOTE:** Local panel owner is enabled when the virtual panel is selected.	Supervisor
Setpoint	CommandSetpoint	Allows you to change the setpoint, if owner = operator.	Supervisor

Engineering Tab

The figure shows an engineering tab:



The table describes the control function of an engineering tab:

Legend	Item name	Description	Security parameter, page 397
Mode			
Mode	Mode	Allows you to view and change the mode. Refer to Operating Mode, page 59.	Set mode - simulation, set mode - maintenance, and Set mode - out of service **NOTE:** Each configurable mode has the different security parameters
Interlock Bypass	InterlockBypass	Interlock Bypass status Allows you to bypass interlock.	Set bypass conditions, set bypass all interlocks

Items

Overview

This section describes the variables, equipstatus and equipstate variable, abnormal and action required conditions, alarm items, events, and trend item of `analog output`.

Variables

The table describes the variable items that are used by Supervision components:

Item Name	Description	Enumeration	Address
Bit items derived from StatusWord.			
ChannelFailure	Output channel inoperable	Healthy/Unhealthy	*Bit0*
InterlockActive	Interlock status	Interlocked/Healthy	*Bit1*
ProgramSetpoint-Mode	Setpoint source for program owner	Remote/Sequence	*Bit2*
Bit items derived from ConfigurationWord.			
OwnerSelect	Owner of control module	Operator/Program	*Bit0*
InterlockBypass	Interlock bypass command	Bypassed/Normal	*Bit1*
Simulation	Simulation mode	–	*Bit13*
Maintenance	Maintenance mode	–	*Bit14*
OutOfService	Analog output unavailable	Out Of Service/In Service	*Bit15*
IO Items			
StatusWord	Status word	–	AOUTPUTGP_ST.STW
ConfigurationWord	Configuration word		AOUTPUTGP_ST. CFGW
Calculated Items			
EquipStatus	Current status of the analog output For more details, refer to EquipStatus calculated variable, page 413.	–	GPL_ AnalogOutputSta-tus
EquipState	Output information For more details, refer to EquipState calculated variable, page 413.		GPL_ AnalogOutputE-quipState
Owner	Current owner of the analog output For more details, refer to calculated variable, page 58.		GPL_OwnerLocal
Mode	Current mode of the analog output Refer to calculated variable, page 59.		GPL_ ModeSignalOutput
Optional Items			
Local panel			
Bit items derived from LPStatusWord.			
LocalPanelMode	Control from field	Enabled/Disabled	*Bit0*
ZeroMode	Analog output control disabled		*Bit1*
ControlSystemMode	Control from remote		*Bit2*
LockedOwner	Owner change disabled from faceplate	Locked/Unlocked	*Bit3*

Item Name	Description	Enumeration	Address
LockedSetpoint	Setpoint change disabled from faceplate		*Bit4*
LocalPanelEnabled	Control from field	Enabled/Disabled	*Bit5*
PushButtonEnabled	Local panel push buttons		*Bit6*
VirtualLocalPanelEnabled	Local panel enable from control		*Bit7*
LocalPanelSelected	Control from field	Selected/Not Selected	*Bit8*
ZeroSelected	Control disabled		*Bit9*
ControlSystemSelected	Control from remote		*Bit10*
PushButtonIncrease	Increase push button signal	On/Off	*Bit11*
PushButtonDecrease	Decrease push button signal		*Bit12*
Bit items derived from LPConfigurationWord.			
VirtualSwitch	Local panel selection from faceplate	Local/Control System	*Bit0*
IO Items			
LPStatusWord	Local panel status word	–	AOUTPUTLPGP_ST.STW
LPConfigurationWord	Local panel configuration word		AOUTPUTLPGP_ST.CFGW
Interlocks	Refer to the topic describing the variable items of interlocks, page 60.		
Maintenance	Refer to the topic describing the variable items of maintenance, page 59.		

EquipStatus Calculated Variable

The table describes the *EquipStatus* calculated variable:

Equipment Status	Conditions	Enumeration
Analog output Closed	Output = MinScale	Closed
Analog output Open	MinScale < Output < FullScale	Open
Analog output Fully Open	Output = FullScale	Fully Open

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	Conditions	Enumeration
Passive	OOS	Close
Active with Maintenance	Not OOS x Maint x OP	Open
Passive with Maintenance	Not OOS x Maint x Not OP	Close
Active with Interlock	Not OOS x Not Maint x ILCK x OP	Open Interlocked
Passive with Interlock	Not OOS x Not Maint x ILCK x Not OP	Close Interlocked
Active	Not OOS x Not Maint x Not ILCK x OP	Open
Passive	Not OOS x Not Maint x Not ILCK x Not OP	Close

Abnormal Conditions

The table describes the abnormal conditions of analog output

Item Name	Description	Address
Bit items derived from Abnormal variable word.		
`AbnormalOwner`	Abnormal owner status appears if the current owner is not a normal owner which configured at the instance level.	*Bit0*
`AbnormalInterlockBy-pass`	Abnormal interlock bypass condition appears if the Interlock bypass is enabled.	*Bit1*
`AbnormalHighHighEnable`	This condition appears, if high-high alarm is not enabled.	*Bit2*
`AbnormalHighEnable`	This condition appears, if high alarm is not enabled.	*Bit3*
`AbnormalMode`	Abnormal mode status appears, if the Mode is not normal.	*Bit4*
`AbnormalLowLowEnable`	This condition appears, if low-low alarm is not enabled.	*Bit5*
`AbnormalLowEnable`	This condition appears, if low alarm is not enabled.	*Bit6*
`AbnormalOperatorTab`	Abnormal status of operator tab. This condition appears if the Interlock bypass is enabled.	*Bit8*
`AbnormalInterlockTab`	Abnormal status of interlock tab.	*Bit11*
`AbnormalEngineeringTab`	Abnormal status of engineering tab.	*Bit12*

Action Required Conditions

The table describes the action required conditions of analog output:

Item Name	Description	Address
Bit items derived from ActionRequired variable word.		
`ActionOperatorTab`	Action required status of operator tab.	*Bit8*
`ActionInterlockTab`	Action required status of interlock tab.	*Bit11*

Alarms

The table describes the alarms that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
`ChannelFailure`	Channel detected error alarm	`GPL_BitValueCheck (StatusWord, 0)`

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item Name	Description
StateAlarm	All changes in the analog output status and output.
ModeAlarm	All changes in the mode of operation of the equipment.
ActionAlarm	All the operator actions in the supervision.

Trends

The table describes the trend items that are used by Supervision components:

Item Name	Expression	Description
Setpoint	AOUTPUTGP_CFG.LSP	Setpoint
Output	AOUTPUTGP_ST.LOP	Output

Digital Input

What's in This Chapter

Template 416
Control 419
Supervision 422

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the digital input.

Template

Overview

This section describes functionality of the `$DigitalInputGP` template.

Description

General Description

The `$DigitalInputGP` process control module template is used to condition a digital signal usually coming from a physical input.

The process template provides timing functions for the connection and/or disconnection, simulation, and can consider the signal status as an alarm function.

Functional Description

The main features of the `$DigitalInputGP` template are described in the following table:

Function	Description
Timing	The input signal is timed to evaluate an output logic signal that absorbs rapid changes in the input signal.
Alarm	You can enable/disable the evaluation of an alarm and indicate the status that needs to be considered for the alarm. You need to incorporate an external logic to evaluate this alarm.
Diagnosis	The template manages the diagnostic status of the signal if the peripherals used provide the signal, and assigns the value that is to be used if the signal is not working properly.
Mode selection	The object switched to different modes of operation like Maintenance and Override.

Optional Services

Available optional service:

- Maintenance

Parameters

Parameters

The `$DigitalInputGP` template provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	**Present value faulty channel state**	Boolean	TRUE	1 = Sets the output signal (*PV*) to 1 when the channel is not operational.
	Watch Alarm	Enum	**Refinement (1)**	Allows enabling (1) or disabling (0) the detected alarm monitoring depending on the dynamic conditions of the process. You can configure the input in 1 of the following ways: • **Interface**: Allows you to connect an interface of type $Bool to the **WatchDI** interface of the instance. • **Refinement**: You can enable/disable the alarm monitoring function during refinement. • **Never**: The alarm monitoring function is disabled. • **Always**: The alarm monitoring function is enabled.
	Present value alarm state		**True Value (3)**	Defines if `PV` value needs to be considered as a detected alarm. You can configure the input in 1 of the following ways: • **Interface**: Allows you to connect an interface of type $Bool to the **WatchDI** interface of the instance. • **Refinement**: You can set the value during refinement. • **False Value**: The function is disabled (0). • **True Value**: The function is enabled (1).
Time	**Set Point for ON Delay**	Duration	00:00:00	Timing set-point for digital input connection `CHIN`) in millisecond (0 to deactivate timing).
	Set Point for OFF Delay			Timing set-point for digital input disconnection `CHIN`) in millisecond (0 to deactivate timing).

Digital Input Signals

The table describes the digital input signal parameters:

Section	Parameter description	Type	De-fault	Additional remarks
Configura-tion	**Negate Signal**	`Boolean`	False	Refer to the topic describing the control DFB, page 419.
	Include Quality Signal			
	Custom Signal Name	`String`	–	

Maintenance

For more details, refer to the topic describing the common parameters of maintenance, page 89.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Security	Set setpoint	`Enum`	Operator (1)	Privilege to change setpoint. This parameter is used in Operator tab, page 424.
	Set mode - override		Supervisor (2)	Privilege to change override mode. This parameter is used in Engineering tab, page 424.
	Set mode - maintenance		Maintenance (3)	Privilege to set maintenance mode. This parameter is used in Engineering tab, page 424.
	Set alarm acknowledge		Operator (1)	Privilege to bypass all interlocks. This parameter is used in Alarm tab, page 67.

Asset and Display Parameters

For more details, refer to the topic describing the common parameters of asset and display, page 43.

Data

The table describes the data parameters:

Section	Parameter description	Type	Default	Additional remarks
Alarm	For more details, refer to the topic describing the common parameters of alarm, page 45.			
Events	Enable Log	`Boolean`	False	Enable event logging for asset status.

Maintenance

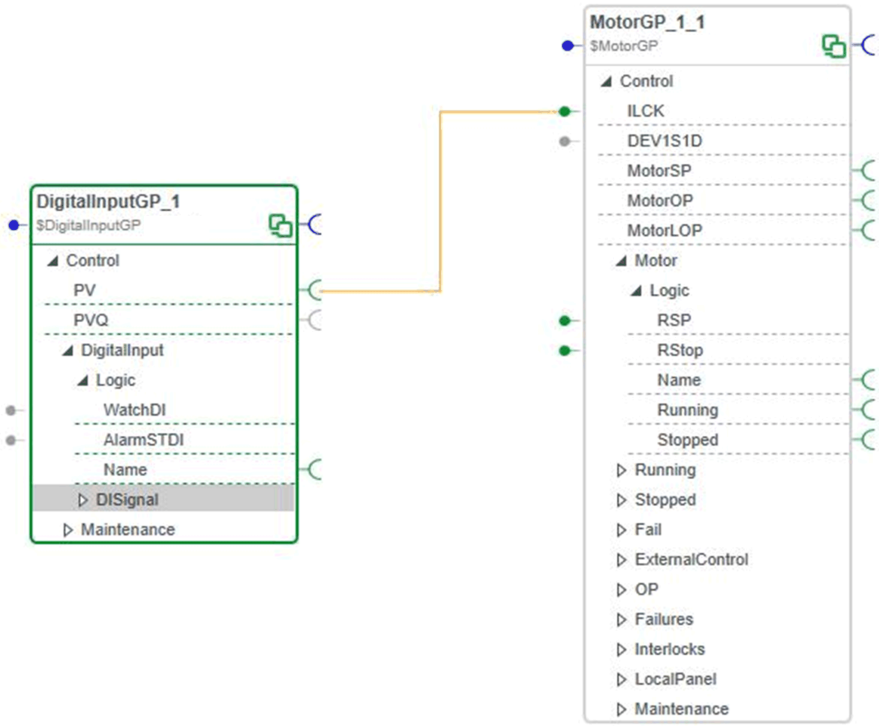
For more details, refer to the topic describing the common parameters of maintenance, page 89.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `DigitalInputGP` (also applicable for `DigitalInputCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



For interface identifiers, refer to the control module template interfaces, page 49.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

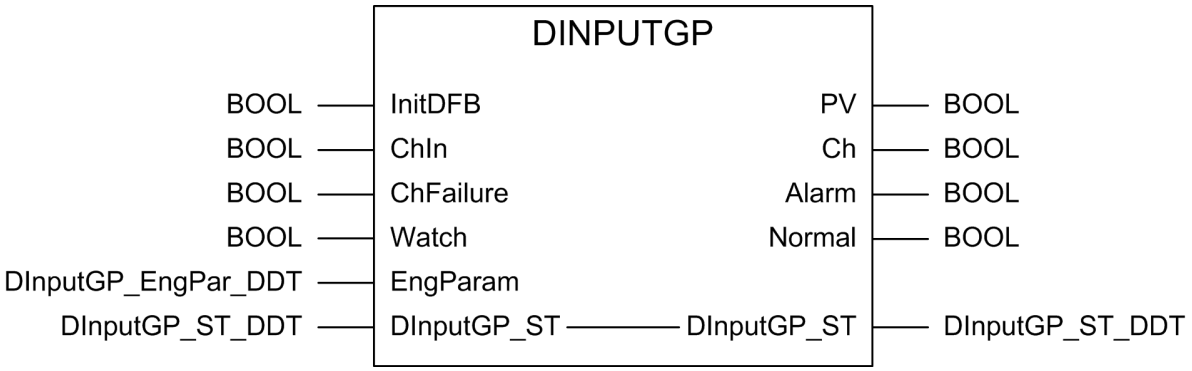
Overview

This chapter describes the `DINPUTGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Name	Type	Description
InitDFB	BOOL	0 = Rising edge is produced on this input, the internal timers of the DFB are reset. If the input and the detected alarm status are timed, the detected alarm and the PV are set to 0.
ChIn	BOOL	1 = Input signal Usually corresponds to a value coming from the input peripherals.
ChFailure	BOOL	1 = Indicates to the DFB whether the input channel that is associated with the *ChIn* input is operational.
Watch	BOOL	1 = Enables the detected alarm monitoring depending on the dynamic conditions of the process.
EngParam	DInputGP_ EngPar_DDT	Engineering parameters.

DInputGP_EngPar_DDT Type

Name	Type	Description
BadPV	BOOL	0 = Value needs to be used (PV output signal and for the purposes of detected alarm evaluation).
AlarmSt	BOOL	1 = PV value needs to be considered as a detected alarm.
TOnSP	TIME	Timing set-point for digital input connection (ChIn).
TOffSP	TIME	Timing set-point for digital input disconnection (ChIn).

Outputs

Output Parameter Description

Name	Type	Description		
PV	BOOL	Timed input value The PV calculation that the DFB performs based on the value of the inputs and the STW input/output is detailed in the following table:		
		STW.Override	**ChFailure**	**PV**
		ON	-	STW.Sim
		OFF	ON	BadPV
		OFF	OFF	Timed ChIn (TOnSP/TOffSP)
Ch	BOOL	Channel value (timers are not considered).		
Alarm	BOOL	1 = Detected alarm evaluated based on the PV value and the DFB configuration is described in the following table:		

STW.AlarmEn	Watch	PV	AlarmSt	Alarm
OFF	-	-	-	OFF
-	OFF	-	-	OFF
ON	ON	OFF	OFF	ON
ON	ON	OFF	ON	OFF
ON	ON	ON	OFF	OFF

Name	Type	Description			
ON		ON	ON	ON	ON
Normal	BOOL	Indicates the equipment operating in normal mode.			

Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
DInputGP_ST	DInputGP_ST_DDT	Provides the DFB status and configuration.

DInputGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	State	*Bit0*	BOOL	1 = Input State is high, refer to output pin PV, page 420.
	AlarmActive	*Bit1*	BOOL	1 = Alarm is detected. Refer to the Alarm output pin, page 420.
	ChFailure	*Bit2*	BOOL	Refer to the ChFailure input pin, page 420.
	AlarmWatch	*Bit3*	BOOL	Indicates if the detected alarm is being monitored. Reproduces the Watch input value.
	AlarmSet-point	*Bit4*	BOOL	PV value needs to be considered as a detected alarm. Reproduces the AlarmSt input value.
	InputValue	*Bit5*	BOOL	Indicates the channel value.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
	AlarmEnable	*Bit0*	BOOL	Enables the detected alarm evaluation to be enabled (1) or disabled (0).
	Override	*Bit1*	BOOL	Refer to Override, page 421.
	OverrideIn-putValue	*Bit2*	BOOL	1 = Override input value.
	Maint	*Bit14*	BOOL	Refer to Maint, page 421.

Public Variables

Public Variable Description

Variable	Type	Description
SC	DInputGP_SC_DDT	Provides the frequently needed data to monitor the discrete input status from the sequential control.

DInputGP_SC_DDT Type

Name	Type	Description
PV	BOOL	Read-only access Refer to the PV output pin, page 420.
Alarm	BOOL	Read-only access Refer to the Alarm output pin, page 420.
Normal	BOOL	Read/write access Refer to the Normal output pin, page 420.
Override	BOOL	Read/write access Enables to set the device to Override mode so that the actual Input signals (ChIn) are ignored. It is simulated that the equipment is in the desired Input (DInputGP_ST.OverrideInputValue).
Maint	BOOL	Read-only access 1 = Maintenance mode is active. Allows the operation of the equipment for maintaining and testing. **NOTE:** Maintenance mode can only change when the Owner is in Operator.

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the management of digital input.

Genies

Genie Properties

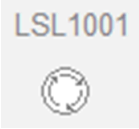
Refer to Genie Properties, page 62.

Genie Icons

Refer to Genie Icons, page 63.

Representation

The table describes the genies of the gpl_digitalinput library:

Graphic symbol	Genie name	Graphic symbol	Genie name
	Circle		*Arrow_Left*
	Arrow_Right		*Emergencystop*
	Pressurelimit		*Switch_Horiz*
	Switch_Vert		*Temperaturelimit*
	Alarmtext		

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

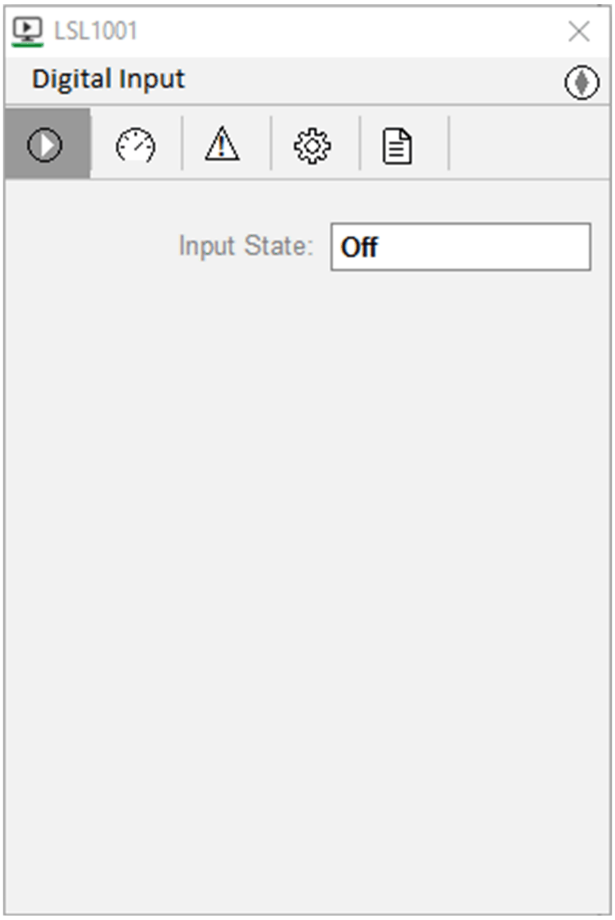
Available Tabs

During operation, clicking an digital input genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
  - Event, page 69
- Tabs for optional functions, which appear only if configured:
  - Measures, page 70

Operator Tab

The figure shows an operator tab:

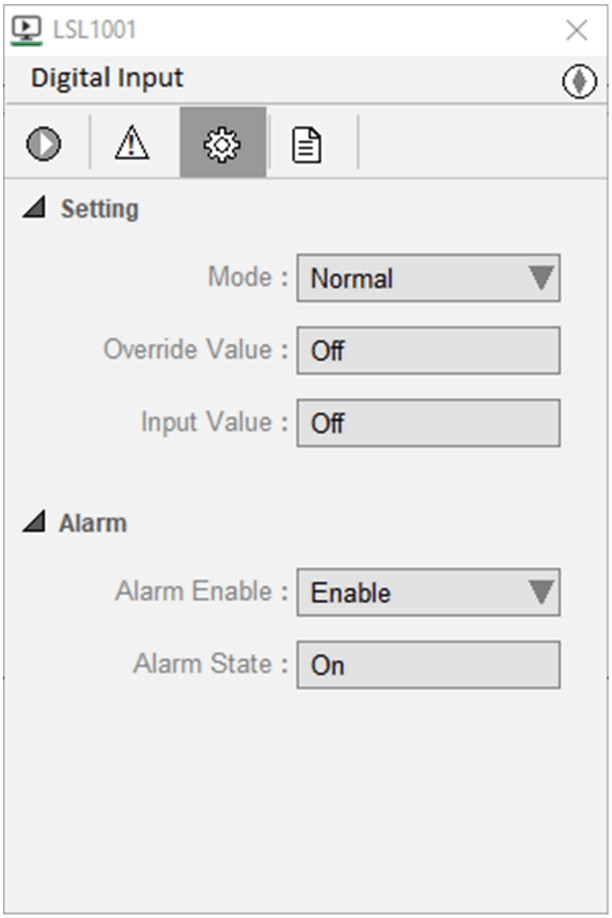


The table describes the control function of an operator tab:

Label	Item Name	Description
Input State	State	Process value

Engineering Tab

The figure shows an engineering tab:



The table describes the control function of engineering tab:

Label	Item name	Description	Security parameter, page 417
Setting			
Mode	Mode	Allows you to view and change the mode. Refer to operating mode, page 59.	Set mode - override, set mode - simulation, set mode - maintenance. **NOTE:** Each configurable mode has the different security parameters
Override Value	OverrideValue	Value will be considered when input source is operator.	Set mode - override
Input Value	ChannelInputValue	Channel value	–
Alarm			
Alarm Enable	AlarmEnable	Alarm monitoring	Set Alarm Acknowledge
Alarm State	AlarmActive	Feedback confirmation detected fault.	Set Alarm Acknowledge

Items

Overview

This section describes the variables, abnormal and action required conditions, alarm items, events, and trends items of `digital input`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
State	Process value	On/Off	*Bit0*
AlarmActive	Feedback confirmation detected fault	Active/Inactive	*Bit1*
ChannelFailure	Input channel inoperable	Healthy/Unhealthy	*Bit2*
AlarmWatch	Alarm watch status	Watched/ Unwatched	*Bit3*
AlarmSetpoint	Value to be considered as alarm	On/Off	*Bit4*
ChannelInputValue	Channel value		*Bit5*
Bit items derived from ConfigurationWord.			
AlarmEnable	Alarm monitoring	Enable/Disable	*Bit0*
InputSource	Input source selection	Operator/Field	*Bit1*
OverrideValue	Value will be considered when input source is operator	On/Off	*Bit2*
Maintenance	Maintenance mode	–	*Bit14*
IO Items			
StatusWord	Status word	–	DINPUTGP_ST.STW
ConfigurationWord	Configuration word		DINPUTGP_ST.CFGW
Calculated Items			
Mode	Current mode of the digital input For more details, refer to calculated variable, page 59.	–	GPL_ ModeSignalInput
Abnormal	Shows abnormal condition of the digital input For more details, refer to Abnormal conditions, page 426.		GPL_ DigitalInputAb-normal
ActionRequired	Shows action required condition of the digital input For more details, refer to action required conditions, page 427.		GPL_ ActionRequired

Abnormal Conditions

The table describes the abnormal conditions of digital input:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
AbnormalAlarmEnable	This condition appears, if High high alarm enable.	*Bit0*
AbnormalMode	Abnormal mode status appears if the current operating mode is not normal mode.	*Bit1*
AbnormalOperatorTab	Abnormal status of operator tab. This condition appears if the Interlock bypass is enabled.	*Bit8*

Item name	Description	Address
`AbnormalFailTab`	Abnormal status of failure tab.	*Bit10*
`AbnormalEngineeringTab`	Abnormal status of engineering tab.	*Bit12*

Action Required Conditions

The table describes the action required conditions of digital input:

Item name	Description	Address
Bit items derived from ActionRequired variable word.		
`ActionOperatorTab`	Action required status of operator tab.	*Bit8*
`ActionFailTab`	Action required status of failure tab.	*Bit10*

Alarms

The table describes the alarms that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
`AlarmActive`	Feedback confirmation fault	`GPL_BitValueCheck(StatusWord, 1)`
`ChannelFailure`	Channel failure alarm	`GPL_BitValueCheck(StatusWord, 2)`

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
`ActionAlarm`	All actions by the operator.

Trends

No trend item is used by Supervision components for `digital input management`.

Digital Output

What's in This Chapter

Template 428
Control 431
Supervision 435

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the digital output.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$DigitalOuputGP` template.

Description

General Description

The `$DigitalOuputGP` process control module template is used to condition a digital signal usually associated with a physical output.

The process template provides interlocking, diagnosis, and owner management functions that you can configure in the Properties window of the instance.

Function Description

The main functions of the digital output template are described in the following table:

Function	Description
Diagnosis	The template manages the diagnostic status of the signal if the periphery used provides it.
Interlocking	The template enables to move to the defined position if an active interlock that requires this move is detected. An interlock bypass function is available.
Owner	Template allows you to configure owner (Operator or Program). As a result, it is responsible for setting the set-point to the desired position.
Mode selection	The object can be switched to different modes of operation like Out Of Service, Simulation and Maintenance.

Optional Services

Available optional services are:

- **Interlocks**
- **Maintenance**

Parameters

Parameters

The `$DigitalOuputGP` template defines operation of a typical digital output. Alternative digital output facets enable the modification of the digital output that are available.

For more details, refer to the topic describing the common parameters of maintenance, page 89.

Control

Control

The table describes the control parameters:

Section	Parameter description	Type	Default value	Additional remarks
Configuration	**Negate signal**	`Boolean`	*FALSE*	–
	Include also quality signal			–
	Userdefined signal conditioning name	`String`	–	Name used for the generated EFB/DFB and variables.

Interlocks

For more details, refer to the topic describing the common parameters of interlocks, page 81.

Maintenance

For more details, refer to the topic describing the parameters of maintenance, page 89.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default value	Additional remarks
Operation	**Normal Owner**	`Enum`	*Program and cascade (5)*	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	**Set owner**	`Enum`	*Operator (1)*	Privilege to change owner. This parameter is used in Operator tab, page 436.
	Set setpoint		*Operator (1)*	Privilege to change setpoint. This parameter is used in Operator tab, page 436.
	Set mode - simulation		*Engineer (5)*	Privilege to change simulation mode. This parameter is used in Engineering tab, page 437.
	Set mode - maintenance		*Maintenance (3)*	Privilege to change maintenance mode. This parameter is used in Engineering tab, page 437.
	Set mode - out of service		*Supervisor (2)*	Privilege to change out of service mode. This

Section	Parameter description	Type	Default value	Additional remarks
				parameter is used in Engineering tab, page 437.
	Set reset		*Operator (confirmed) (10)*	Privilege to reset. This parameter is used in Operator tab, page 436.
	Set alarm acknowledge		*Operator (1)*	Privilege to acknowledge alarm. This parameter is used in Alarm Tab, page 67.
	Set bypass conditions		*Supervisor (2)*	Privilege to set bypass conditions mode. This parameter is used in Engineering tab, page 437.
	Set bypass all interlocks		*Supervisor (2)*	Privilege to bypass all interlocks. This parameter is used in Engineering tab, page 437.
	Set asset maintenance		*Operator (1)*	Privilege to set asset maintenance mode. This parameter is used in Engineering tab, page 437.
Events	**Enable Log**	`Boolean`	*FALSE*	Enable event logging for asset status.

Asset and Display Parameters

For more details, refer to the topic describing the common parameters of asset and display, page 43.

Alarm Parameters

For more details, refer to the topic describing the common parameters of alarm, page 45.

Interlocks

For more details, refer to the topic describing the common parameters of interlocks, page 81.

Maintenance

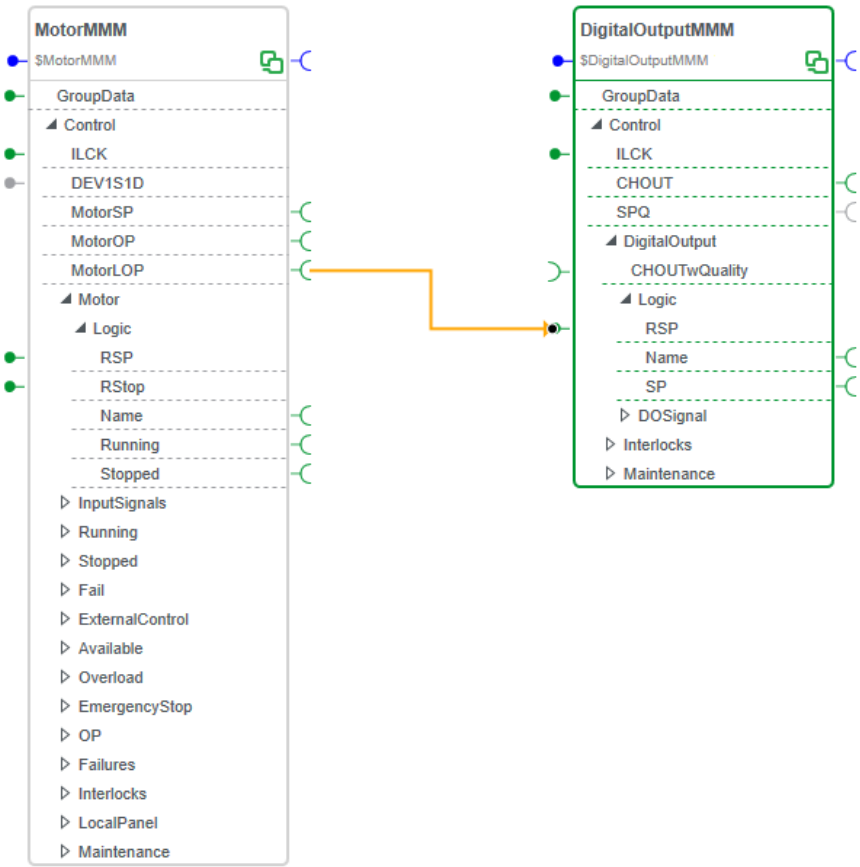
For more details, refer to the topic describing the parameters of maintenance, page 89.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

The following figure depicts the `$DigitalOuputGP` (also applicable for `$DigitalOuputCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



The table describes the control module template interfaces:

Interface identifier	Type/Role	Description	Example
`CHOUT`	*$Bool/Def*	Channel output value of output interface to other template.	Channel output value or setpoint connect as a channel input to digital input template.
`SP`	*$Bool/Def*	Current output interface to other template.	–

For interface identifiers, refer to the control module template interfaces, page 49.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

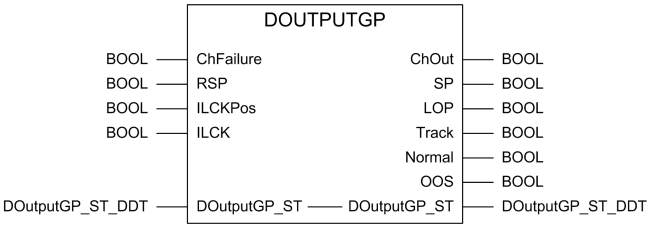
Overview

This chapter describes the `DOUTPUTGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
ChFailure	BOOL	1 = Indicates to the DFB whether the output channel associated with the ChOut output is operational.
RSP	BOOL	1 = Remote set-point. Usually set by the continuous control; for example, by a PID controller output.
ILCKPos	BOOL	Position during Interlocked (1 = Activate output in case if interlock is active).
ILCK	BOOL	1 = Interlocks the device at the defined position.

Outputs

Output Parameter Description

Parameter	Type	Description		
ChOut	BOOL	1= Field output signal active. During simulation the signal will be deactivated. The output calculation that the DFB performs is detailed in the following table:		
		SC.ILCKD		**ChOut**
		ON		ILCKPos
		OFF		SP
SP	BOOL	1 = Current set-point. The output calculation that the DFB performs is detailed in the following table:		
		Owner (OFF: Program, ON: Operator)	**Rem**	**SP**
		OFF	OFF	SC.LSP
		OFF	ON	RSP
		ON	-	DOutputGP_CFGW.LSP
LOP	BOOL	1 = Logical output signal active.		
Track	BOOL	1 = Determines that the remote set-point (RSP input) is one being considered for the ChOut output calculation. This output is usually used to report to the algorithm connected to the RSP input (that is, a PID) whether the control loop is open (Track is 1) or not (Track is 0) so that it can be repositioned based on the output set-point that is actually in effect. The output calculation that the DFB performs is detailed in the following table:		

Parameter	Type	Description				
		ILCKD	**Owner**	**Rem**	**OOS**	**Track** is calculated as:
		-	-	-		OFF
		ON	-	-	OFF	ON
		OFF	OFF	OFF	OFF	ON
		OFF	OFF	OFF	OFF	OFF
		OFF	ON	-	OFF	ON
Normal	BOOL	Indicates the equipment operating in normal mode.				
OOS	BOOL	1 = Out of service mode is active.				

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
DOutputGP_ST	DOutputGP_ST_DDT	Provides the DFB status and configuration.

DOutputGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	ChFailure	*Bit0*	BOOL	Refer to the ChFailure input pin, page 432.
	ILCKD	*Bit1*	BOOL	Refer to ILCKD, page 434.
	Rem	*Bit2*	BOOL	Allows the DFB to be configured to remote set-point—RSP—(1) or local set-point—LSP— (0).
	LOP	*Bit3*	BOOL	Refer to the LOP output pin, page 432.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
	Owner	*Bit0*	BOOL	Read/write access Enables to configure whether the set-point is set by the Program (0) or the Operator (1).
	ILCKBP	*Bit1*	BOOL	Read/write access Allows the interlock to be bypassed (1).
	SP	*Bit2*	BOOL	Refer to the SP output pin, page 432.
	Sim	*Bit13*	BOOL	Refer to Sim, page 434.
	Maint	*Bit14*	BOOL	Refer to Maint, page 434.
	OOS	*Bit15*	BOOL	Refer to the OOS output pin, page 432.

Public Variables

Public Variable Description

Variable	Type	Description
SC	DOutputGP_SC_DDT	Provides the frequently needed data to monitor and control the analog output status from the sequential control.

DOutputGP_SC_DDT Type

Name	Type	Description
LSP	REAL	Read/write access Allows the assignment of the local set-point for the sequential control if the owner is the Program (Owner input/output is set to 0) and the selected set-point is the Local on (SC.Rem public variable is set to 0). Otherwise, the current set-point (SP output) is continuously copied to this variable.
SP	REAL	Read-only access Refer to the SP output pin, page 432.
LOP	REAL	Read-only access Refer to the LOP output pin, page 432.
Rem	BOOL	Read/write access Refer to the Rem inputs/outputs pin, page 433.
Track	BOOL	Read-only access Refer to the Track output pin, page 432.
BadSt	BOOL	Read-only access Indicates whether the input channel is operational or not. Reproduces the value of the ChFailure input value.
ILCKD	BOOL	Read-only access The signal evaluation depending on the ILCK input and the DOutputGP_ST.CFGW.ILCKBP input/output is shown in the following table: **ILCK** / **ILCKBP** / **SC.ILCKD** is calculated as: OFF / - / OFF ON / OFF / ON ON / ON / OFF
Owner	BOOL	Read/write access Refer to the Owner inputs/outputs pin, page 433.
Normal	BOOL	Read/write access Refer to the Normal output pin, page 432.
Sim	BOOL	Read/write access Enables to set the device to simulation mode so that output pin OP get suppressed. If LOP = 1, then it is not possible to change the simulation mode.
Maint	BOOL	Read-only access Refer to the Maint inputs/outputs pin, page 433.
OOS	BOOL	Read/write access 1 = Maintenance mode is active. Allows the operation of the equipment for maintaining and testing. Interlock conditions are bypassed.

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the management of digital output.

Genies

Genie Properties

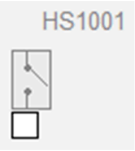
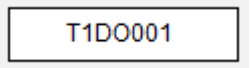
Refer to Genie Properties, page 62.

Genie Icons

Refer to Genie Icons, page 63.

Representation

The table describes the genies of the `gpl_digitaloutput` library:

Graphic symbol	Genie name
	Circle
	Switch_Horiz
	Siren
	*Switch_Vert*
	*doinstanceid_m*

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

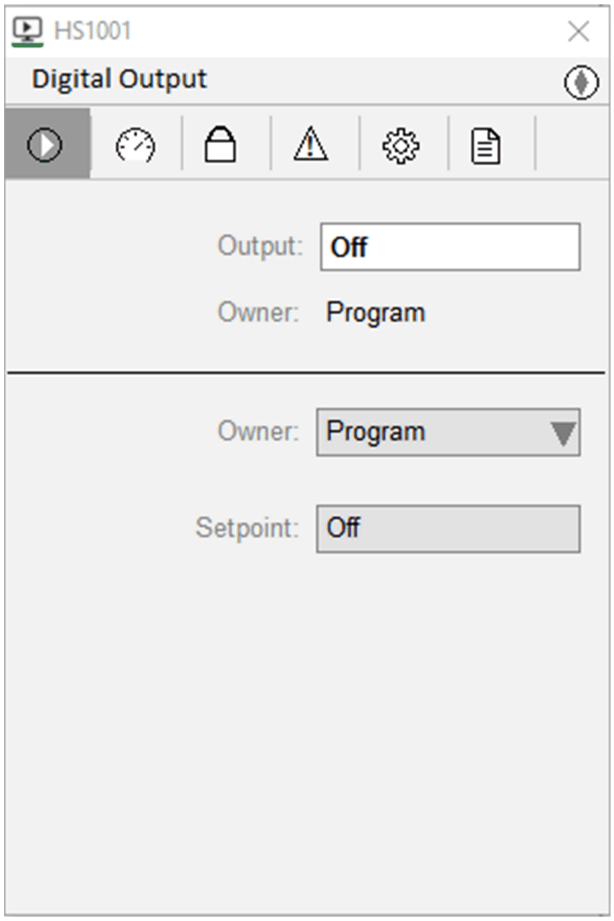
Available Tabs

During operation, clicking a digital output genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
  - Event, page 69
- Tabs for optional functions, which appear only if configured:
  - Interlock, page 87
  - Measures, page 70

Operator Tab

The figure shows an operator tab:

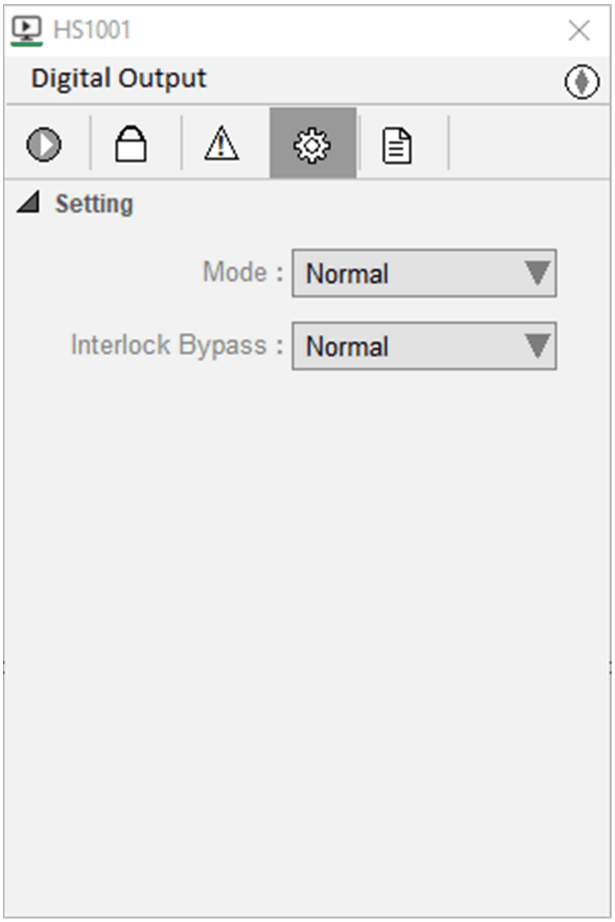


The table describes the control function of an operator tab:

Label	Item name	Description	Security parameter, page 429
Output	Output	Output status of digital output. For more details, refer to EquipState calculated variable, page 159.	—
Owner	Owner	Current owner of analog output, For more details, refer to operating owner, page 58.	
Owner	OwnerSelect	Owner command of analog output (Program or Operator).	Set owner
Setpoint	Setpoint	Allows you to change the setpoint, if owner = operator.	Set Setpoint

Engineering Tab

The figure shows an engineering tab:



The table describes the control function of engineering tab:

Label	Item name	Description	Security parameter, page 429
Mode	Mode	Allows you to view and change the mode. For more details, refer to operating mode, page 59.	Set mode - normal, set mode - override, set mode - simulation, set mode - maintenance, and Set mode - out of service **NOTE:** Each configurable mode has the different security parameters
Interlock Bypass	InterlockBypass	Interlock bypass status Allows you to bypass interlock.	Set bypass conditions, set bypass all interlocks

Items

Overview

This section describes the variables, equipstate variable, abnormal and action required conditions, alarm items, events, and trend item of `digital output`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`ChannelFailure`	Output channel inoperable	Healthy/Unhealthy	*Bit0*
`InterlockActive`	Interlock status	Interlocked/Healthy	*Bit1*
`ProgramSetpointMode`	Setpoint source for program owner	Remote/Sequence	*Bit2*
`Output`	Output status	On/Off	*Bit3*
Bit items derived from ConfigurationWord.			
`OwnerSelect`	Owner of control module	Operator/Program	*Bit0*
`InterlockBypass`	Interlock bypass command	Bypassed/Normal	*Bit1*
`Setpoint`	Desired state of output	On/Off	*Bit2*
`Simulation`	Simulation mode	–	*Bit13*
`Maintenance`	Maintenance mode		*Bit14*
`OutOfService`	Digital output unavailable	Out Of Service/In Service	*Bit15*
IO Items			
`StatusWord`	Status word	–	`DOUTPUTGP_ST.STW`
`ConfigurationWord`	Configuration word		`DOUTPUTGP_ST.CFGW`
Calculated Items			
`Owner`	Current owner of the digital output Refer to calculated variable, page 58.		`GPL_Owner`

Item name	Description	Enumeration	Address
Mode	Current mode of the digital output Refer to calculated variable, page 59.		GPL_ ModeSignalOutput
Abnormal	Shows abnormal condition of the digital output Refer to Abnormal Conditions, page 439.		GPL_ DigitalOutputAb-normal
ActionRequired	Shows action required condition of the digital output Refer to action required conditions, page 440.		GPL_ ActionRequired
Optional Items			
Interlocks	Refer to the topic describing the variable items of interlocks, page 60.		
Maintenance	Refer to the topic describing the variable items of maintenance, page 59.		

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	Conditions	Enumeration
Passive with Maintenance	OOS x Maint	Off
Active with Maintenance	Not OOS x Maint x OP	On
Passive	Not OOS x Maint x Not OP	Off
Active with Interlock	Not OOS x Not Maint x ILCK x OP	On Interlocked
Passive with Interlock	Not OOS x Not Maint x ILCK x Not OP	Off Interlocked
Active	Not OOS x Not Maint x Not ILCK x OP	On
Passive	Not OOS x Not Maint x Not ILCK x Not OP	Off

Abnormal Conditions

The table describes the abnormal conditions of digital output:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
AbnormalOwner	Abnormal owner status appears, if the current owner is not a normal owner which configured at the instance level.	*Bit0*
AbnormalInterlockBy-pass	Abnormal interlock bypass condition appears, if the Interlock bypass is enabled.	*Bit1*
AbnormalMode	Abnormal mode status appears if the current operating mode is not normal mode.	*Bit2*
AbnormalOperatorTab	Abnormal status of operator tab. This condition appears, if the Interlock bypass is enabled.	*Bit8*
AbnormalInterlockTab	Abnormal status of interlock tab.	*Bit11*
AbnormalEngineeringTab	Abnormal status of engineering tab.	*Bit12*

Action Required Conditions

The table describes the action required conditions of digital output:

Item name	Description	Address
Bit items derived from ActionRequired variable word.		
`ActionOperatorTab`	Action required status of operator tab.	*Bit8*
`ActionInterlockTab`	Action required status of interlock tab.	*Bit11*

Alarms

The table describes the advanced alarm items that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expression
`ChannelFai-lure`	Output channel detected error	`GPL_BitValueCheck (StatusWord, 0)`

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
`ActionAlarm`	All the operator actions in the supervision.

Trends

No trend item is used by Supervision components for `digital output`.

Range

What's in This Chapter

Template 441

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the range.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$RangeGP` template.

Description

General Description

The `$RangeGP` process control module template is used to configure range values for variables such as `PV`, `SP`, and so on.

This process template helps to configure parameters such as high value, low value, engineering unit, format, and memory address.

Parameters

Parameters

The `$RangeGP` template defines operation of analog range.

The table describes the range parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Low range (EU)	`Float`	0.0	–
	High range (EU)		100.0	–
	Engineering units	`String`	%	–
	Format		%###.#	–

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$RangeGP` (also applicable for `$RangeCE`) template as it appears in `Links Editor` and an example of other templates connected to it:

The table describes the control module template interfaces:

Interface identifier	Type/Role	Description	Example
`AnalongRange`	*$Range/Def*	–	

Totalizer

What's in This Chapter

Template 443
Control 447
Supervision 455

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the totalizer.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$TotalGP` template.

Description

General Description

The objective of `$TotalGP` template is to perform totalization calculation based on 3 inputs (analog signal, digital pulses, counter).

Functional Description

The main features of the `$TotalGP` template are described in the following table:

Function	Description
`Control`	Controls start and stop of totalization of the input from field equipment/ instrument.
`Owner selection`	The template manages the control system level, as to which is the owner (operator or program). As a result, it is responsible for setting the setpoint for the desired position.
`Totalizing`	The totalization is done using trapezoidal rule algorithm for analog inputs, while it is summation for digital as well as counter values.
`Monitoring`	Enables the user to monitor the totalized value, last totalized value, setpoint and active operation state.
`Processing states and Commands`	The template processes the commands received from Supervision function and determines the active operation state.
`Local and remote setpoint`	Enables to monitor totalized value with the setpoint that is determined through a setpoint selector (local or remote). The local setpoint is assigned to the sequential control or to command received from the monitoring system. At the same time, remote setpoint is assigned to DFB control from the logic implemented in continuous control.

Parameters

Parameters

The `$TotalGP` template provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Setpoint	Force remote setpoint	`Boolean`	False	Enables the remote control operation for totalizer.
Configuration	Totalization co-efficient	`Float`	1.0	Totalized conversion coefficient applied to totalizing calculations. This is one time configuration.
	Auto start on idle	`Boolean`	False	Whether the operation to be started automatically once reaching `Idle` state.
	Auto reset on stop			Whether the operation to be reset automatically once reaching `Stopped` state.
Configuration flow	Sample rate	`Duration`	00:00:01	Time interval at which sample is considered for totalization.
Configuration counter	Rollover maximum count	`Unsigned long`	0	Enter the maximum roll over value of counter module.
Failures	For more details, refer to the topic describing the parameters of failures, page 74.			
Counter signal				
Configuration	Custom signal name	`String`	–	Instance name
	Select counter channel data type	`Enum`	UDINT (0)	The data type of the counter value.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Operation	Normal owner	`Enum`	Program & Cascade (5)	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	Set owner	`Enum`	Operator (1)	Privilege to change owner. This parameter is used in Operator Tab, page 457.
	Set setpoint		Operator (1)	Privilege to change setpoint. This parameter is used in Operator Tab, page 457.
	Set reset		Operator (Confirmed) (10)	Privilege to reset. This parameter is used in Operator Tab, page 457.
	Set parameters		Engineer (5)	Privilege to set parameters. This parameter is used in Engineering Tab, page 458.
	Set alarm acknowledge		Operator (1)	Privilege to acknowledge alarm. This parameter is used in Alarm Tab, page 67.
	Set Bypass conditions		Supervisor (2)	Privilege to bypass condition. This parameter is used in Interlock Tab, page 87.
	Set start		Operator (1)	Privilege to set start. This parameter is used in Operator Tab, page 457.
	Set hold		Operator (1)	Privilege to set hold. This parameter is used in Operator Tab, page 457.
	Set restart		Operator (1)	Privilege to set restart. This parameter is used in Operator Tab, page 457.

Section	Parameter description	Type	Default	Additional remarks
	Set stop		Operator (1)	Privilege to set stop. This parameter is used in Operator Tab, page 457.
	Set reset		Operator (1)	Privilege to reset. This parameter is used in Operator Tab, page 457.
Asset and Display Parameters	For more details, refer to the topic describing the parameters of asset and display, page 43.			
Trend	For more details, refer to the topic describing the parameters of trend, page 47.			

Data

The table describes the data parameters:

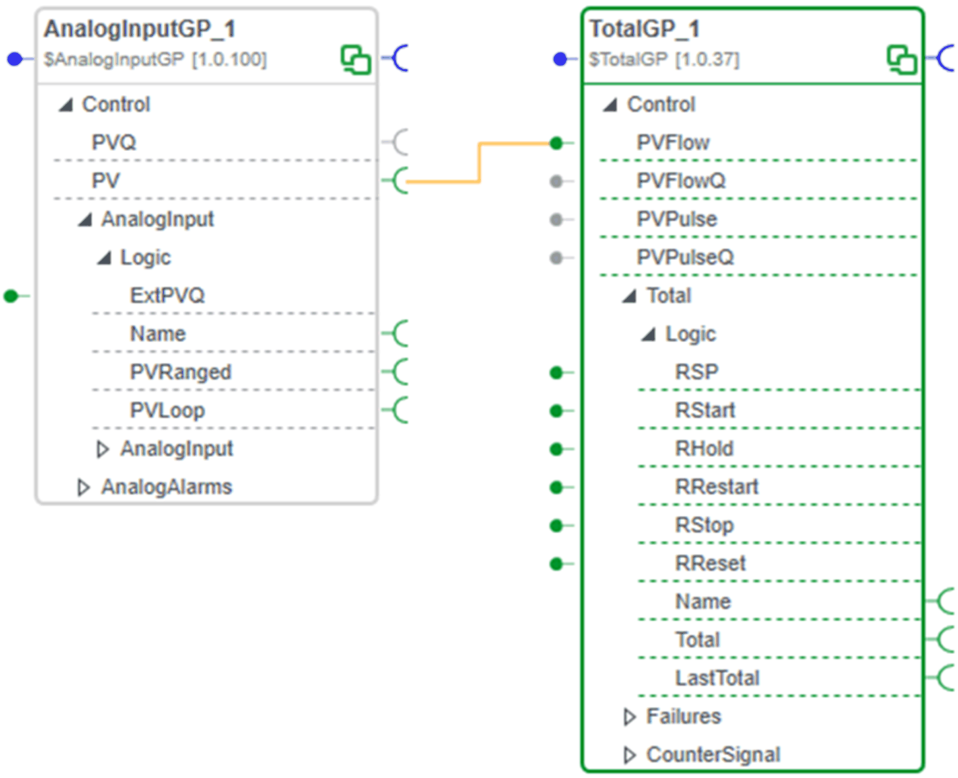
Section	Parameter description	Type	Default	Additional remarks
Range	Optimal range	`Enum`	Disable (0)	You can configure the optimal range: • **Disable**: Optimal range limits • **Fixed Value** - Configure optimal limits as per requirement. • **Setpoint Deviation** - Optimal range calculation based on Setpoint and Setpoint deviation.
	Practical range disable	`Boolean`	True	–
	Engineering unit	`String`	%	The format type to be used in the supervision.
	Format type		####.#EU	The format type to be used in the supervision.
Alarm	Held with error description	`String`	@(`failure` detected while totalizing)	–
	Held with error severity	`Enum`	Low (4000)	–
	Held with error group	`String`	`Failure`	–
Operation	Normal state	`Enum`	Idle & Running (4)	–
	Disable setpoint	`Boolean`	–	It disables setpoint tag.
Disable Buttons	start, stop, hold, reset, and restart command button	`Boolean`	False	–
Deadband	Total value	`Float`	0.0	–
Events	Enable log	`Boolean`	False	Enable event logging for asset status.
Trend	Disable total	`Boolean`	False	It disables total trend tag.
	Disable setpoint		False	It disables setpoint trend tag.
Historize	Setpoint	`Boolean`	False	It allows to historize the change in setpoint.
	Total/Last total			It allows to historize the change in total or last total.
	Optimal range high/low			It allows to historize the change in optimal high or optimal low range.
Failures	For more details, refer to the topic describing the parameters of failures, page 74.			

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$TotalGP` (also applicable for `$TotalCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



The following link interface create the link symbol on faceplate:

- PVFlow or PVFlowQ

For more details, refer to Supervision chapter > Faceplate > Navigation Link of General Purpose Library User Guide.

The table describes the control module template interfaces:

Interface identifier	Type/Role	Description	Example
`PVFlow`	*$Real/Ref*	Process value flow.	–
`PVFlowQ`	*$ASignalwithQuality/ Ref*	Process value flow with detected fail. Ref nested interface consisting of `$Real` and `$Bool`	–
`PVPULSE`	*$Bool/Ref*	Process value pulse.	–
`PVPulseQ`	*$DSignalwithQuality/ Ref*	Process value pulse with detected fail. Ref nested interface consisting of `$Real` and `$Bool`	–
`Rhold`	*$Bool/Ref*	Rising edge of hold command.	–
`RRestart`	*$Bool/Ref*	Rising edge of restart command.	–

Interface identifier	Type/Role	Description	Example
RReset	*$Bool/Ref*	Rising edge of reset command.	–
Name	*$TotalIFGP/Def*	Instance name concatenated with control block name.	–
Total	*$Real/Ref*	Present totalized value in engineering units.	–
Lasttotal	*$Real/Ref*	Last totalized value in engineering units	–
CounterSignalUDINT	*$UDInt/Ref*	External logical variable of UDINT data type.	–
CounterSignalUINTExt	*$CNTRSignalwQuality/Ref*	External logical variable of UINT data type.	–
CounterSignalUDINT	*$UInt/Ref*	Counter input signal external with quality.	–
CounterSignalUINTExt	*$CNTRSignalwQuality/Ref*	Counter input signal external with quality.	–

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

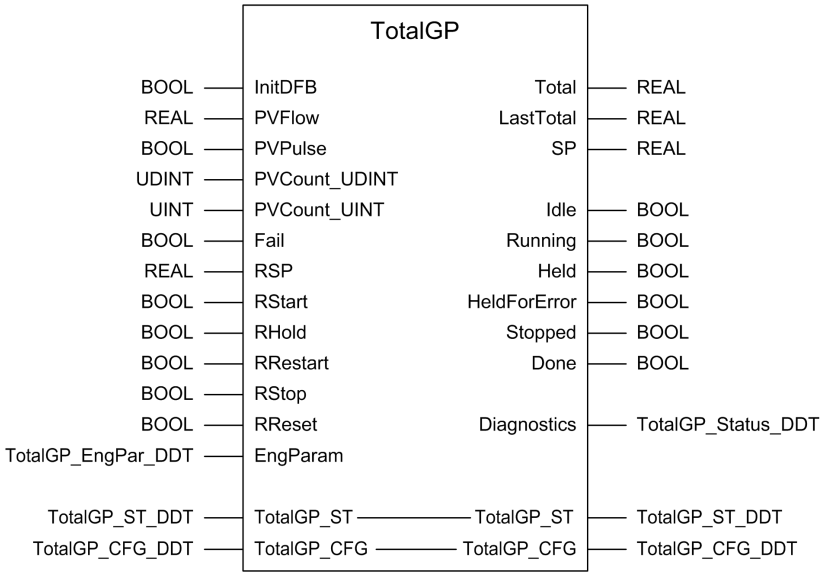
Overview

This chapter describes the `TotalGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller



Inputs

Input Parameter Description

Parameter	Type	Description
InitDFB	*BOOL*	Rising edge on this input resets DFB internal timers, data, etc. **NOTE:** • The input is not reset (0) by the DFB. • The reference value is updated with the latest value.
PVFlow	*REAL*	Process flow value of the measurement in engineering units. **NOTE:** • It is mandatory to connect the input pin to either a variable or other blocks to use this DFB in this mode. • Negative values of PV are not considered for totalization.
PVPulse	*BOOL*	This input pin accepts series of pulses sent from the field equipment/ instrument. **NOTE:** • Only rising edge is considered. • It is mandatory to connect the input pin to either a variable or other blocks to use this DFB in this mode.
PVCount_Udint	*UDINT*	This input pin accepts a count value of an unsigned double integer data type sent from the field equipment/instrument. **NOTE:** • Only upcounter is supported. • It is mandatory to connect the input pin to, either a variable or other blocks to use this DFB in this mode. • If *RolloverCount* is not declared then the rollover count is considered to be 4294967295 in this mode.
PVCount_Uint	*UINT*	This input pin accepts a count value of an unsigned integer data type sent from the field equipment/instrument. **NOTE:** • Only upcounter is supported. • It is mandatory to connect the input pin to, either a variable or link to other blocks to use this DFB in this mode. • If *RolloverCount* is not declared then the rollover count is considered to be 65535 in this mode.
Fail	*BOOL*	A detected error in the device/ module is indicated to the DFB. **NOTE:** When channel becomes healthy, *Total* starts to run automatically
RSP	*REAL*	This setpoint will be considered for processing *Done* status. The remote set point is considered only if the below conditions are satisfied: • *SC.Rem* = 1 • Owner = Program **NOTE:** The range of setpoint is: • Lower value: 0.0 • Higher value: 999999999.9999999
RStart	*BOOL*	A rising edge sends a start command only when program/remote-cascade mode is configured.
RHold	*BOOL*	A rising edge sends a hold command only when program/remote-cascade mode is configured.
RRestart	*BOOL*	A rising edge sends a restart command only when program/remote-cascade mode is configured.
RStop	*BOOL*	A rising edge sends a stop command only when program/remote-cascade mode is configured.
RReset	*BOOL*	A rising edge sends a reset command only when program/remote-cascade mode is configured.
RolloverCount	*UDINT*	The rollover to be declared as the maximum PV value provided to the input pin of *PVCount_Udint* or *PVCount_Uint*.

Parame-ter	Type	Description
		NOTE: • It is mandatory to connect the input pin to, either a variable or link to other blocks to use this DFB in this mode. • If *RolloverCount* is not declared then ◦ if *PVCount_Udint* is selected then rollover count is considered to be 4294967295. ◦ If *PVCount_Uint* is selected then rollover count is considered to be 65535. • If *RolloverCount* is not declared, also if *PVCount_Udint* is not selected or if *PVCount_Uint* is not selected then rollover count is set to 0. • If input value is greater than rollover count then totalizing is deemed to be valid. Although rollover is not considered after this scenario. • In case the counter is restarted, wherein rollover is invalid, then user has to verify that the rising edge is triggered to *InitDFB* to reinitialize totalizing operation before providing new input to the DFB. The total value is not affected during reinitialize. For example, Case 1: If rollover count is 10. The total when input value changes from 0 to 5 is, 5. Continuing, when input value changes from 5 to 9, the total will be 9. Further, when the input value rolls back from 9 to1, the total will be 11. Case 2: If rollover count is not configured and *PVCount_Uint* is selected. The total when input value changes from 0 to 5 is, 5. Continuing, when input value changes from 5 to 9, the total will be 9. Further, when the input value rolls back from 9 to 2, the total will be 65537.
K	*REAL*	Conversion factor is applied to the totalizing calculation output. This allows the total to be calculated in the desired engineering units. **NOTE:** • Constant *K* is one time configuration. • *K* is considered only when the DFB has a transition from *Idle* state to *Running* state. • Negative integers are not valid. • If K=0. Then, the default value taken will be 1.0. For example, Case 1 • *ScanRate* = 1s • Constant flow rate = 1 Litre per hour at 9.00 AM. Totalized value in Litres at 11.00 AM is as below. AuxTotal is considered to be internal variable. Total is the output of DFB. AuxTotal = 7200L × second / hr. *K* to be configured as (second / hour) which turns out to be, *K* =(1/3600) = 0.00027 Total = AuxTotal × *K* Total = 7200 × 0.00027 Total = 2 Litres. Case 2 • Scanrate = 500ms • Constant Flow rate = 1 Litre per hour at 9.00 AM. User wants to see totalized value in Litres at 11.00 AM. AuxTotal is considered to be internal variable. Total is the output of DFB. AuxTotal = 14400 [(L× 500ms)/hr.] *K* to be configured as (millisecond/hour). [(500 × 0.001)/3600] = 0.000139 Total = AuxTotal × *K*

Parame-ter	Type	Description
		Total = 14400 × 0.000139 Total=2 Litres.
Scan-Rate	*TIME*	Only applicable for *PVFlow*. Allows the user to configure sample acquisition time. The DFB takes a sample each time, a time equal to or greater than the one configured for this input has elapsed. For calculation purposes, the time elapsed between one DFB execution and another is taken into account. **NOTE:** If no value provided, then default value T#1s is considered as *ScanRate*. User need to verify that the DFB execution parameters are compatible with the time configured in *ScanRate* parameter; this allows the DFB to run with a high frequency to carry out sampling with the configured time including latency time. In any case, the DFB considers any possible execution delays regarding the theoretical time adjusted in this parameter when it comes to calculations.
AutoS-tart	*BOOL*	When *Idle* state is reached, the function acts as it is receiving a *Start* command automatically (without consuming any extra execution cycle). After *Start* command, the operation will move to *Running* state.
AutoRe-set	*BOOL*	When *Stopped* state is reached, the function acts as it is receiving a *Reset* command automatically (without consuming any extra execution cycle). After *Reset* command, the operation will move to *Idle* state

NOTE: Priorities are set based on the inputs that are selected for totalization.

Default priorities set for the inputs:

- *PVFlow* = 0
- *PVPulse* = 1
- *PVCount_Udint* = 2
- *PVCount_Uint* = 3

Lower the value, higher the priority, that is, 0 is the maximum priority. If more than one input is connected, detected configuration error occurs. During detected configuration error, totalization continues without any interruption based on the priorities of inputs that are set. Only during *Idle* state the user can change the input priorities.

Outputs

Output Parameter Description

Parameter	Type	Description
Total	REAL	Present totalized value in engineering units. Refer Formula for Totalization topic for additional information. **NOTE:** *Total* is reset to 0 when reset command given.
LastTotal	REAL	Holds the last *Total* value when reset command is executed.
SP	REAL	Current set point.
Idle	BOOL	Inactive operation. Waiting for *Start* command to start totalizer operation.
Running	BOOL	*Running* state is reached after the *Start* command is processed. Totalizer operation starts calculating.
Held	BOOL	*Held* state is reached after the *Hold* command is processed. Totalizer operation is paused.
HeldForError	BOOL	*HeldForError* state appears when detected failure occurs during *Running* state, due to which totalizer operation is paused. **NOTE:** Totalizer operation runs automatically, if a detected failure does not occurs during *HeldForError* state considering latest PV as reference PV.
Stopped	BOOL	*Stopped* state appears after *Stop* command is processed. Requires an initialization command (*Reset*) to go to inactive (*Idle*) state.

Parameter	Type	Description
Done	BOOL	*Done* state will be high when present total value reaches *SP* (setpoint). Despite *Done* state being high, *Total* output will keep calculating/ totalizing. **NOTE:** *Done* state is not applicable when setpoint is 0.
Diagnostics	*TotalStatusGP_DDT*	Structure reporting the detected errors and status of the DFB.
	Parameter	**Description**
	KApplied	Value of *KApplied* is captured for calculation when the state of DFB is changed from *Idle* state to *Running* state.
Diagnostics	*Info*	A word with a series of bits reporting the detected errors and status of the DFB. **NOTE:** In case the user requires an alarm to be generated for conditions in the *Info* word, the bit representing the detected error can be connected to the *CONDSUMGP* DFB, which in turn is connected to the *Fail* input of *TotalGP* DFB. • Bit 0: Detected configuration error. **NOTE:** This condition is true when: More than one input PV or no PV, which are *PVFlow*, *PVPulse*, *PVCount_Udint* and *PVCount_Uint* are connected. *PVCount_Udint*, *PVCount_Uint* value is more than the configured value of *RolloverCount*. • Bit 1: Input *K* is not valid. **NOTE:** ◦ Range of *K* which can be configured is 0 to 1000 inclusive. ◦ *KNotValid* is checked only during *Running* or *Held* or *HeldForError* state. ◦ PV value is not added to total, when the value of *K* is out of range. • Bit 2: Rollover flag. **NOTE:** True of bit represents *Total* has crossed 999999999.9999999 The totalizing operation is reset automatically. • Bit 3: *PVFlow* selected for totalization. • Bit 4: *PVPulse* selected for totalization. • Bit 5: *PVCount_Udint* selected for totalization. • Bit 6: *PVCount_Uint* selected for totalization. • Bit 7: Negative *PVFlow* value. **NOTE:** PV value is not added to *TOTAL*, when *PVFlow* value is negative.

Formula for Totalization

The Control function offers totalization of 3 type of inputs:

- Analog (*REAL*) - based on trapezoidal rule

  Aux = (b-a) × [f(b)+f(a)]/2

  AuxTotal = AuxTotal + Aux

  Total = AuxTotal × K
- Digital Pulses (*BOOL*)

  AuxTotal = AuxTotal + [Rising edge (Digital input)]

  Total = AuxTotal × K
- Counter (*LONG*)

  AuxTotal = AuxTotal + (Current - Last)]

  Total = AuxTotal × K

Where:

b = Time in which the current sample is taken.

a = Time in which the previous sample was taken.

f(b) = Value of the current measurement.

f(a) = Value of the previous measurement.

NOTE: AuxTotal and Aux variables are used for internal calculation.

SP Calculation

Owner	SC.REM	Output SP is calculated as
Program	0 = OFF	*SC.LSP* (Local Set Point)
Program	1 = ON	*SetPoint* (Remote Setpoint from input pin of *TotalGP* DFB)
Operator	-	HMI/ faceplate of *TotalGP* function block

Inputs/Outputs

Inputs/Outputs Parameter Description

Parameter	Type	Description
TotalGP_ST	*TotalGP_ST_DDT*	Data structure holds the minimum information required for performing controlling and monitoring functions. The information used by the operator screen is readable and writable from the HMI/ SCADA system.
TotalGP_CFG	*TotalGP_CFG_DDT*	Provides the data necessary to configure the DFB (normally from the monitoring subsystem).

TotalGP_ST_DDT Type

Parameter	Type	Description
State	*INT*	Provides active state of the operation. Read-only access to the data is contained in the integer. State info: • State = 1 (*Idle*) • State = 2 (*Running*) • State = 4 (*Held*) • State = 8 (*HeldForError*) • State = 16 (*Stopped*)
Command	*INT*	Enable commands to perform operation from the Supervision system. Command info: • Command = 1 (Start) • Command = 2 (Hold) • Command = 4 (Restart) • Command = 8 (Stop) • Command = 16 (Reset)
STW	*WORD*	Status Word. • Bit 0: REM (0= Local; 1=Remote). • Bit 1: *Done* • Bit 2: Alarm (*HeldForError*)

Parameter	Type	Description
		• Bit 3: Rollover flag. Refer *Info* parameter in Output Parameter Description topic (see Modicon Libraries General Purpose, Process Components User Guide). • Bit 4: *AutoStart* • Bit 5: *AutoReset* • Bit 6: *FailActive*
CFGW	*WORD*	Provides the data needed to configure the DFB (normally from the monitoring system). Enables to configure whether the setpoint is set by: • Bit 0: Owner (0= Program; 1=Operator) • Bit 3: Reset rollover flag
Total	*REAL*	Present totalized value in engineering units. Refer Formula for Totalization topic for additional information (see Modicon Libraries General Purpose, Process Components User Guide).
TotalINT	*UDINT*	Integer part of *Total* in engineering units.
TotalDEC	*REAL*	Decimal part of *Total* in engineering units.
SP	*REAL*	This variable indicates what the current setpoint is, and is calculated internally in the DFB as long as the owner is not the operator. When owner is operator, the setpoint is configured from the HMI system. **NOTE:** The range of setpoint is: • Lower value: 0.0 • Higher value: 999999999.9999999

TotalGP_CFG_DDT Type

Parameter	Type	Description
LastTotal	*REAL*	The value on current total in engineering units when it was reset.
LastTotalINT	*UDINT*	Integer part of *LastTotal* in engineering units.
LastTotalDEC	*REAL*	Decimal part of *LastTotal* in engineering units.

Public Variables

Public Variable Description

Parameter	Type	Description
SC	*TotalGP_SC_DDT*	Provides the frequently needed data for monitoring and controlling of the DFB.

*TotalGP_SC_DDT*Type

Parameter	Type	Description
LSP	*REAL*	Enables to assign the local target setpoint for the sequential control if the program is the owner (Owner input/output = OFF) and the selected set point is local (*SC.REM* public variable = OFF). Otherwise, the current target setpoint is continuously copied to this variable. **NOTE:** The range of setpoint is: • Lower value: 0.0 • Higher value: 999999999.9999999
Total	*REAL*	Present totalized value in engineering units. Refer Formula for Totalization topic for additional information (see Modicon Libraries General Purpose, Process Components User Guide). **NOTE:** *Total* is reset to 0 when reset command given.
Properties	*WORD*	*WORD* bits are expressed as *RANKED* bit Bit 0 (Rem): • 1 = Allows the DFB to consider commands from program remote mode. • 0 = Allows the DFB to consider commands from program local mode Bit 1 (*Owner*): Provides the data needed to configure the DFB normally from the monitoring subsystem. Enables to configure whether the setpoint is set by: • 0 = Program • 1 = Operator Bit 2 (*Start*): Command to start the operation of the *TotalGP* DFB from *Idle* to *Running* state. It is only valid if the state is inactive (*Idle*). Bit 3 (*Hold*): Command to hold the operation of the *TotalGP* DFB from *Running* to *Held* or *HeldForError* to *Held* state. Bit 4 (*Restart*): Command to restart the operation of *TotalGP* DFB from *Held* to *Running* state. Bit 5 (*Stop*): Command to stop the operation of *TotalGP* DFB from *Running* to *Stopped* or *Held* to *Stopped* state. Bit 6 (*RESET*): Command to reset the operation of *TotalGP* DFB from *Stopped* to *Idle* State. Bit 7 (*Idle*): Inactive operation. Waiting for *START* command to start totalizer operation. Bit 8 (*Running*): *Running* state appears after the *Start* command is processed. Totalizer operation starts calculating. Bit 9 (*Held*): *Held* state is reached after the *Hold* command is processed. Totalizer operation is paused. Bit 10 (*HeldForError*): *HeldForError* state appears when detected failure occurs during *Running* state, due to which totalizer operation is paused. Bit 11 (*Stopped*): *Stopped* state is reached after *Stop* command is processed. Requires an initialization command (*RESET*) to go to inactive (*IdleIdle*) state. Bit 12 (*CONFIGURATIONERROR*): If more than one input is connected, detected configuration error occurs. Bit 13 (*Done*): *Done* state will be high when present Total value reaches *SP* (setpoint). Total output will keep calculating/ Totalizing. *Done* state is not applicable when setpoint is 0.

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the management of totalizer.

Genies

Genie Properties

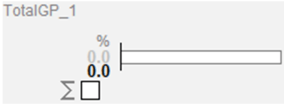
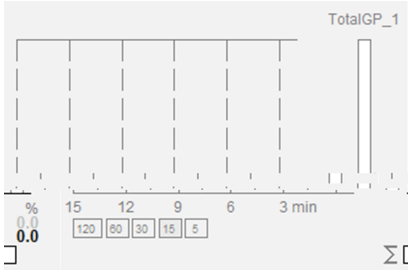
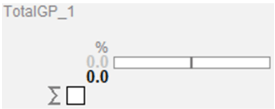
Refer to Genie Properties, page 62.

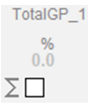
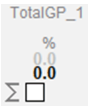
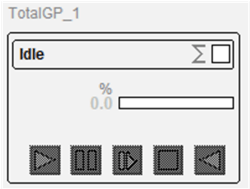
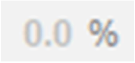
Genie Icons

Refer to Genie Icons, page 63.

Representation

The table describes the genies of the `gpl_total` library:

Graphic symbol	Genie name	Description
	*Bar_TotalSP_Horiz*	Setpoint (SP) status display on a horizontal bar with engineering units.
	Bar_TotalSP_Vert	Setpoint (SP) status display on a vertical bar with engineering units.
	*Bar_TotalSP_Vert_Trend*	Setpoint (SP) status display on a vertical bar with engineering units. In addition, the symbol displays trends and allows you to select the trend period in minutes.
	*Dev_TotalSP_Horiz*	Setpoint (SP) deviation display on a horizontal bar with engineering units.

Graphic symbol	Genie name	Description
	Dev_TotalSP_Vert	Setpoint (SP) deviation display on a vertical bar with engineering units.
	*Indicator_Total*	Total value indicator
	*Indicator_TotalSP*	Total value and setpoint indicator
	*PanelWithTotal*	Panel with total
	PanelWithTotalSP	Panel with total setpoint
	Total	Total value
	*TotalWithEU*	Total value with engineering units

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

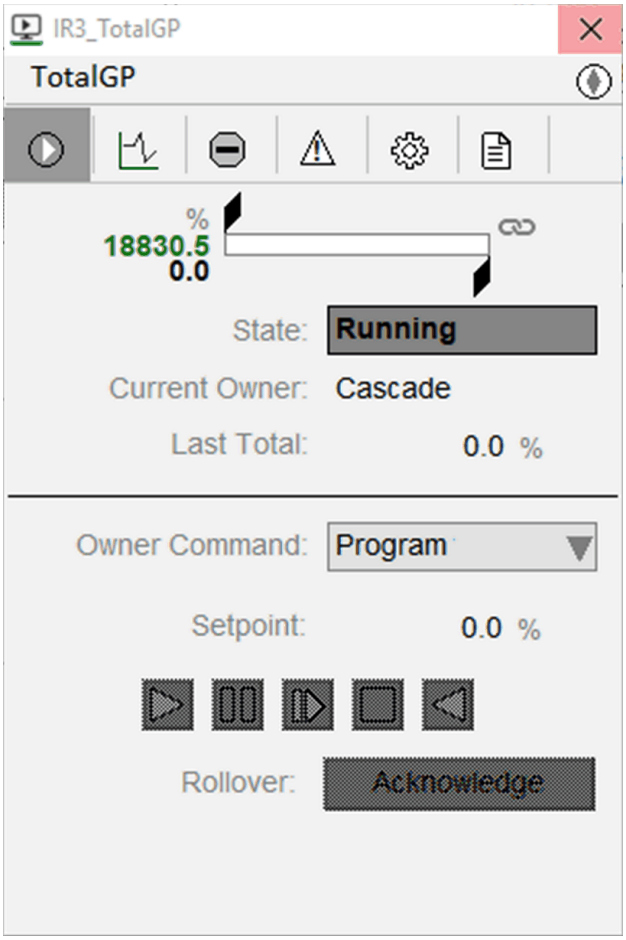
Available Tabs

During operation, clicking a total genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
  - Event, page 69
  - Trend, page 69
- Tabs for optional functions, which appear only if configured:
  - Failure, page 78

Operator Tab

The figure shows an operator tab:

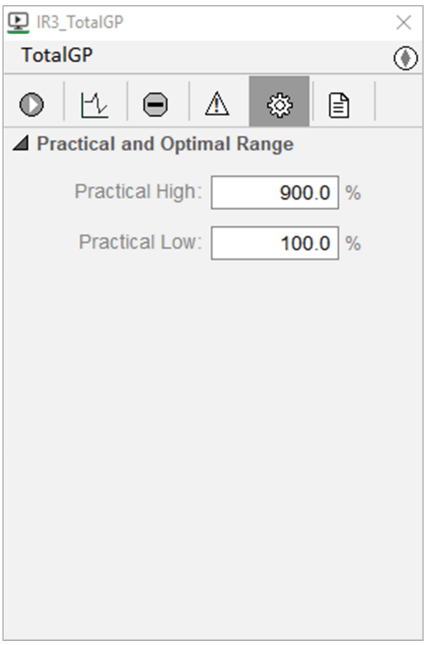


The table describes the control function of an operator tab:

Label	Item name	Description	Security parameter, page 444
Output	Output	Output status of totalizer. For more details, refer to EquipState calculated variable, page 460.	—
Current Owner	Owner	Current owner of totalizer, For more details, refer to operating owner, page 58.	—
Last Total	LastTotal	The last total value in engineering units when reset command is executed.	—
Owner Command	OwnerCommand	Owner command of totalizer (Program or Operator).	Set owner
Setpoint	Setpoint	Allows you to change the setpoint, if owner = operator.	Set Setpoint

Engineering Tab

The figure shows an Engineering tab:



The table describes the control function of an engineering tab:

Label	Item name	Description	Security parameter, page 444
Practical and Optimal Range			
Practical High	PRHigh	Practical range high.	—
Practical Low	PRLow	Practical range low.	—

Items

Overview

This section describes the variables, EquipStatus and equipstate variable, abnormal and action required conditions, alarm items, events, and trends item of `totalizer`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`ProgramSetpointMode`	Setpoint source for program owner	Remote/Sequence	*Bit0*
`Done`	Setpoint equal current total	–	*Bit1*
`HeldwithError`	Detected failure	–	*Bit2*
`RolloverFlag`	Total value has crossed 999999999	–	*Bit3*
`AutoStart`	Status of auto start	–	*Bit4*
`AutoReset`	Status of auto reset	–	*Bit5*

Item name	Description	Enumeration	Address
FailActive	Detected failure	Active/Inactive	*Bit6*
Bit items derived from ConfigurationWord.			
OwnerSelect	Owner of control module	Operator/Program	*Bit0*
ResetFlag	Command to reset the rollover flag	–	*Bit3*
Bit items derived from Command.			
Start	Start totalizing	–	*Bit0*
Hold	Hold totalizing	–	*Bit1*
Restart	Restart totalizing	–	*Bit2*
Stop	Stop totalizing	–	*Bit3*
Reset	Reset totalizing	–	*Bit4*
IO Items			
StatusWord	Status word	–	TotalGP_ST.STW
ConfigurationWord	Configuration word		TotalGP_ST.CFGW
Command	Commands to control the operation	–	TotalGP_ST. Command
Total	Totalized value	–	TotalGP_ST.Total
SP	Setpoint	–	TotalGP_ST.SP
LastTotal	Last totalized value	–	TotalGP_ST. LastTotal
PRHigh	Practical high	–	TotalGP_PRHigh
PRLow	Practical low	–	TotalGP_PRLow
DeviationSetpoint	Deviation setpoint	–	TotalGP_ DeviationSetpoint
DeviationHighSet-point	Optimal deviation high	–	TotalGP_ DeviationHighSet-point
DeviationLowSet-point	Optimal deviation low	–	TotalGP_ DeviationLowSet-point
LastTotalINT	Integral part of last totalized value	–	TotalGP_ST. LastTotalINT
LastTotalDEC	Decimal part of last totalized value	–	TotalGP_ST. LastTotalDEC
TotalINT	TotalGP_ST. TotalINT	–	Integral part of totalized value
TotalDEC	Decimal part of totalized value	–	TotalGP_ST. TotalDEC
PR	Practical range	–	TotalGP_PR
Calculated Items			
Owner	Current owner of the totalizer Refer to calculated variable, page 58.		GPL_Owner
EquipState	Current state of the totalizer. Refer to Equipstate calculated variable, page 460.		GPL_ TotalizerState
Equipstatus	Current status of the totalizer.		GPL_ TotalizerStatus

Item name	Description	Enumeration	Address
	Refer to EquipStatus calculated variable, page 460.		
`Abnormal`	Shows abnormal condition of the totalizer Refer to Abnormal Conditions, page 460.		`GPL_Abnormal`
`ActionRequired`	Shows action required condition of the totalizer Refer to action required conditions, page 461.		`GPL_ ActionRequired`
`ORHigh`	Optimal range high		`GPL_ORHighCalc, GPL_ ORHighAbsolute-Calc`
`ORLow`	Optimal range low		`GPL_ORLowCalc, GPL_ ORLowAbsoluteCalc`
`ORDsp`	Optimal range display		`True`
Optional Items			
Failures	Refer to the topic describing the variable items of failures, page 61.		

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	Conditions	Enumeration
Active	`Running` x Not `HeldwithError`	Running
Active with error	`Running` x `HeldwithError`	Held
Passive	(`Idle` + `Held` + `HeldForError` + `Stopped`) x Not[HeldwithError	Stopped
Passive with error	(`Idle` + `Held` + `HeldForError` + `Stopped`) x [HeldwithError	Held for error

EquipStatus Calculated Variable

The table describes the *EquipStatus* calculated variable:

Equipment State	Conditions	Enumeration
Totalizing done	`Done` x Not `RolloverFlag`	Totalizing Done
Rollover	Not `Done` x `RolloverFlag`	Rollover
Totalizing Done and rollover	`Done` x `RolloverFlag`	Totalizing Done and rollover

Abnormal Conditions

The table describes the abnormal conditions:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
`AbnormalOwner`	Abnormal owner status appears, if the current owner is not a normal owner which configured at the instance level.	*Bit0*
`AbnormalState`	Abnormal state	*Bit3*
`AbnormalOperatorTab`	Abnormal status of operator tab. This condition appears, if the Interlock bypass is enabled.	*Bit8*
`AbnormalFailTab`	Abnormal status of `failure` tab.	*Bit10*

Action Required Conditions

The table describes the action required conditions:

Item name	Description	Address
Bit items derived from ActionRequired variable word.		
`ActionOperatorTab`	Action required status of operator tab.	*Bit8*
`ActionFailTab`	Action required status of `failure` tab.	*Bit10*

Alarms

The table describes the advanced alarm items that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expression
`HeldwithEr-ror`	Failure detected while totalizing	`GPL_BitValueCheck (StatusWord, 2)`

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
`StateAlarm`	All changes in the anaolog input status and output.
`ModeAlarm`	All changes in the mode and owner.
`ActionAlarm`	All actions by the operator.

Trends

The table describes the trend items that are used by Supervision components:

Item name	Expression	Description
`PV`	`TotalGP_ST.Total`	Present value
`SP`	`TotalGP_ST.SP`	Setpoint
`ORHigh`	`TotalGP_ORHigh`	Optimal range high
`ORLow`	`TotalGP_ORLow`	Optimal range low

Multiple Analog Input

What's in This Chapter

Template 462
Control 466
Supervision 472

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the multiple analog input.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$AnalogInMultiGP` template.

Description

General

The $AnalogInMultiGP process control module template is used to condition up to 4 analog signals normally coming from physical inputs and to select one of them based on the chosen selection criterion. The range can be configured from the Supervision runtime.

The process template provides linear scaling, cut-off, diagnosis, and simulation functions that you can configure in the Properties window of the instance.

Functional Description

The main features of the `$AnalogInMultiGP` template are described in the following table:

Function	Description
`Scaling`	The template scales the input signal (usually in raw data) to engineering units through a linear function.
`Cut-Off`	You can configure a minimum value, which is called the cut-off value. If the measured value received from the transmitter is below the cut-off value, the measured value is non-useable. In such case, the cut-off value is being considered.
`Diagnosis`	The template manages the diagnostic status of the signal if the peripherals used provide the signal, and assigns the value that is to be used if the signal is not working properly.
`Mode selection`	The object can be switched to different modes of operation like Maintenance and Override.
`External PV`	The template enables you to connect a signal that is already in engineering units (and therefore does not need scaling, diagnosis, or cut-off operations) while maintaining the simulation function.
`Optimal range`	You can configure the optimal range based on the defined range or Setpoint deviation.

Optional Services

Available optional services are:

- Limit Alarm
- Deviation Alarm
- Rate Alarm

Parameters

Parameters

The `$AnalogInMultiGP` template provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Range	Parameters are same as the `$AnalogInputGP`, refer to control, page 379.			
Setpoint	Force remote setpoint	`Boolean`	False	Enables the remote control operation for multi analog input.
Time	Setpoint for connecting the alarms	`Duration`	00.00.03	–
	Setpoint for disconnecting the alarms	`Duration`	00.00.05	–

Analog Input Signals

The table describes the analog input signal parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Include quality signal (applicable for 'Int' data type)	`Boolean`	False	Refer to the topic describing the control DFB, page 466.
	Custom signal name	`String`	–	

Analog Alarms

For parameters of analog alarms, refer to parameter, page 109.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Trend	For more details, refer to the topic describing the parameters of trend, page 47.			
Security	Set mode - override	`Enum`	Supervisor (2)	Privilege to change override mode. This parameter is used in Engineering Tab, page 477.
	Set reset		Operator (Confirmed) (10)	Privilege to reset. This parameter is used in Operator Tab, page 475.

Section	Parameter description	Type	Default	Additional remarks
	Set Owner		Operator (1)	Privilege to change owner. This parameter is used in Operator Tab, page 475.
	Set setpoint		Operator (1)	Privilege to change setpoint. This parameter is used in Operator Tab, page 475.
	Set alarm limits		Operator (1)	Privilege to set alarm limits. This parameter is used in Operator Tab, page 475.
	Set alarm acknowledge		Operator (1)	Privilege to acknowledge alarm. This parameter is used in Alarm Tab, page 67.
	Set mode - maintenance		Maintenance (3)	Privilege to set maintenance mode. This parameter is used in Engineering Tab, page 477.
	Set parameters		Engineer (5)	Privilege to set parameters. This parameter is used in Operator Tab, page 475.
Operation	Normal owner	`Enum`	Program & cascade (5)	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Asset and Display Parameters	For more details, refer to the topic describing the parameters of asset and display, page 43.			

Data

The table describes the data parameters:

Section	Parameter description	Type	Default	Additional remarks
Range	Optimal range	`Enum`	Disable (0)	You can configure the optimal range: • **Disable**: Optimal range limits • **Fixed Value** - Configure optimal limits as per requirement. • **Setpoint Deviation** - Optimal range calculation based on Setpoint and Setpoint deviation.
	Practical range high/low disable	`Boolean`	True	It disables practical range high and practical low value.
	Practical range disable			It disables the practical range indication.
Alarm	Selected channel `failure` description	`String`	@(Selected channel `failure` alarm)	A detail description for alarms.
	Selected channel `failure` severity	`Enum`	Critical (1000)	It allows to set the criticality (Critical (1000), High (2000), Medium (3000) and Low (4000)) of alarm.
	Selected channel `failure` group	`String`	Channel	Group name for rapid alarm filtering.
Alarm **NOTE:** There	Channel {x} `failure` description	`String`	@(Channel{x} `failure` alarm)	A detail description for alarms.

Section	Parameter description	Type	Default	Additional remarks
are 4 channels are used in multi analog input. {x} represents from 1 to 4.	Channel {x} `failure` severity	`Enum`	Critical (1000)	It allows to set the criticality (Critical (1000), High (2000), Medium (3000) and Low (4000)) of alarm.
	Channel {x} `failure` group	`String`	Channel	Group name for rapid alarm filtering.
	Sensor deviation description	`String`	@(Sensor deviation alarm)	
	Sensor deviation severity	`Enum`	High (2000)	
	Sensor deviation group	`String`	Alarm	
Disable	Present value Bad	`Boolean`	False	–
	Present value Cutoff			–
	Override value			–
Deadband	Present value	`Integer`	0	Deadband value for present value.
Events	Enable log	`Boolean`	False	It allows to historize the change setpoints.
Trend	Disable present value, present value 1, present value 2, present value 3, and present value 4,	`Boolean`	False	–
Historize	PV/PV1/PV2/PV3/PV4,	`Boolean`	False	It allows to historize the change in PV/PV1/PV2/PV3/PV4.
	Ranges			It allows to historize the change in ranges.
	Input selector			It allows to historize the change in input selector.

For parameters of analog alarms, refer to parameter, page 109.

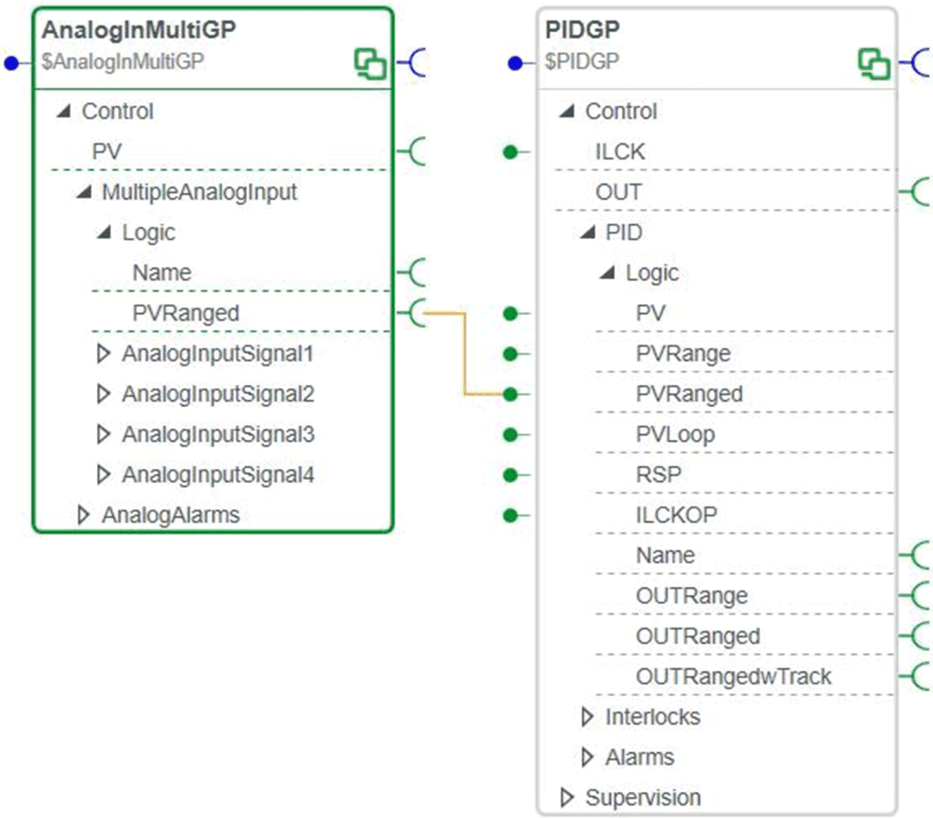
Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

Interfaces

This figure shows the `$AnalogInMultiGP` (also applicable for `$AnalogInMultiCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



The following link interface create the link symbol on faceplate:

- PV

The table describes the control module template interfaces:

Interface identifier	Type/Role	Description	Example
`AINPUTSignalExt` **NOTE:** This interface is applicable for analog signals (AnalogInputSignal1 to AnalogInputSignal4).	*$AINPUTSignal/HO*	Analog input conditioning signal external.	–
`Name`	*$MAInputIFGP/Def*	Instance name concatenated with control block name.	–

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

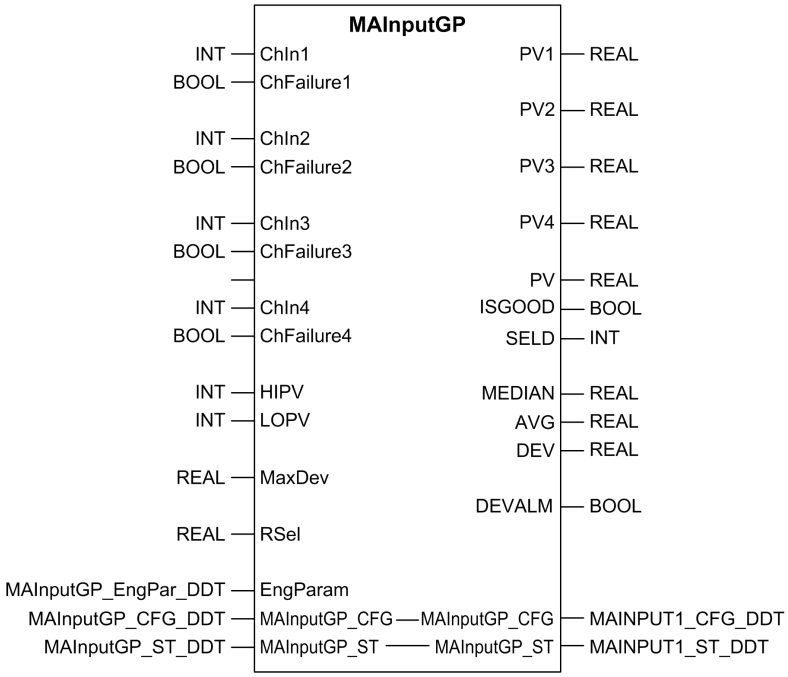
Overview

This chapter describes the `MAInputGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
ChIn1	INT	Channel 1 input signal. Normally corresponds to a value in raw data coming from the input/output peripherals.
ChFailure1	BOOL	1 = Indicates to the DFB that the input channel associated with the ChIn1 input is operational.
ChIn2	INT	Channel 2 input signal. Normally corresponds to a value in raw data coming from the input/output peripherals.
ChFailure2	BOOL	1 = Indicates to the DFB that the input channel associated with the ChIn2 input is operational.
ChIn3	INT	Channel 3 input signal. Normally corresponds to a value in raw data coming from the input/output peripherals.
ChFailure3	BOOL	1 = Indicates to the DFB that the input channel associated with the ChIn3 input is operational.
ChIn4	INT	Channel 4 input signal. Normally corresponds to a value in raw data coming from the input/output peripherals.
ChFailure4	BOOL	1 = Indicates to the DFB that the input channel associated with the ChIn4 input is operational.
HIPV	REAL	Maximum permissible value of the high range of measurement in engineering units (PV output) corresponding to the maximum value (configured in the HIRaw input) of the ChIn1 to ChIn4 inputs. Used as the high range by default if the high range has not been modified from the monitoring subsystem (MAInputGP_CFG.CHIPV).
LOPV	REAL	Minimum permissible value of the low range of measurement in engineering units (PV output) corresponding to the minimum value (configured in the LORaw input) of the ChIn1 to ChIn4 inputs. Used as the low range by default if the low range has not been modified from the monitoring subsystem (MAInputGP_CFG.CLOPV).
MaxDev	REAL	Maximum permissible standard deviation for showing a detected alarm.
RSel	INT	Enables to select the selection method as shown in the following table:
		RSel — **PV** is calculated as:

Parameter	Type	Description	
		0	First value of the possible PVx signals that are enabled and without a detected fault. (PV1 has maximum priority, PV4 has minimum priority).
		1	PV1
		2	PV2
		3	PV3
		4	PV4
		5	Median of signals enabled and without a detected fault.
		6	Average value of signals enabled and without a detected fault.
		7	Minimum value among the enabled PVx signals.
		8	Maximum value among the enabled PVx signals.
HIRaw	INT	High range of the ChIn1 to ChIn4 input signals.	
LORaw	INT	Low range of the ChIn1 to ChIn4 input signals.	
TONSP	TIME	Time for connecting the detected alarm due to standard deviation.	
TOFFSP	TIME	Time for disconnecting the detected alarm due to standard deviation.	

Outputs

Output Parameter Description

Parameter	Type	Description		
PV1..PV4	REAL	Calculated measurement value normally in engineering units. The DFB calculates the PVx measurement based on the value of the inputs and of the MAInpuGP_ST and MAInpuGP_CFG inputs/outputs. PVx measurement is shown in the following table:		
		ChFailurex	**ChInx**	**PVx** is calculated as:
		ON	-	MAInpuGP_CFG.BadPV
		OFF	<= LORaw	MAInpuGP_CFG.CLOPV
		OFF	> LORaw and < HIRaw	Refer to the formula.
		OFF	>= HIRaw	MAInpuGP_CFG.CHIPV
PV	REAL	Value of the output (PVx) selected with the input signal.		
IsGood	BOOL	1 = Indicates the following: • quality of the current selection • selection is correct • channel is functional • simulation mode is inactive		
SELD	INT	Selected selection method.		
Median	REAL	Median of PVx signals enabled and without a detected fault.		
Avg	REAL	Average of PVx signals enabled and without a detected fault.		
Dev	REAL	Standard deviation of the channels that are enabled and without a detected fault.		
DevAlm	BOOL	1 = The standard deviation is higher than the maximum value specified with the MaxDev input.		

Linear Scaling formula:

$$\left(\frac{(CHIPV - CLOPV)}{(HiRaw - LORaw)} \times ChInx\right) + CLOPV - \left(\frac{(CHIPV - CLOPV)}{(HiRaw - LORaw)} \times LORaw\right)$$

If the resulting value of the calculation is less than `MAInpuGP_CFG.CutOffPV` (and the signal comes from the `ChInx` input), the `PVx` output is set to match the `MAInpuGP_CFG.CLOPV` input/output.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
`MAInpuGP_ST`	`MAInpuGP_ST_DDT`	Provides the necessary data for monitoring the DFB status and configuring its simulation when required.
`MAInpuGP_CFG`	`MAInpuGP_CFG_DDT`	Provides the data needed to configure the DFB normally from the monitoring subsystem.

`MAInpuGP_ST_DDT` Type

Name	Type	Description
`STW`	`WORD`	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
`CFGW`	`WORD`	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
`PV`	`REAL`	Read-only access Refer to the `PV` output pin, page 468.

`MAInpuGP_CFG_DDT` Type

Name	Type	Description
`PV1`	`REAL`	Read-only access Refer to the `PV1` output pin, page 468.
`PV2`	`REAL`	Read-only access Refer to the `PV2` output pin, page 468.
`PV3`	`REAL`	Read-only access Refer to the `PV3` output pin, page 468.
`PV4`	`REAL`	Read-only access Refer to the `PV4` output pin, page 468.
`CHIPV`	`REAL`	High range of the measurement in engineering units (`PVx` outputs) corresponding to the maximum value (configured in the `HIRaw` input) of the `ChInx` inputs. The maximum value that this variable can assume is limited by the value of the `HIPV` input. If the range entered from the monitoring system is incorrect (`MAInpuGP_CFG.CLOPV` = `MAInpuGP_CFG.CHIPV` = 0.0), it is initialized with the value of the `HIPV` input.

Name	Type	Description
CLOPV	REAL	Low range of the measurement in engineering units (PVx outputs) corresponding to the minimum value (configured in the HIRaw input) of the ChInx inputs. The minimum value that this variable can assume is limited by the value of the LOPV input. If the range entered from the monitoring system is incorrect (MAInpuGP_CFG.CLOPV = MAInpuGP_CFG.CHIPV = 0.0), it is initialized with the value of the LOPV input.
BadPV	REAL	Value needs to be used in engineering units (PVx output signals) when the channel is not operational. The value that this variable can assume is limited by the range configured in engineering units (MAInpuGP_CFG.CLOPV and MAInpuGP_CFG.CHIPV). If the range entered from the monitoring system is incorrect (MAInpuGP_CFG.CLOPV = MAInpuGP_CFG.CHIPV = 0.0), it is initialized with the value of the HIPV input.
CutOffPV	REAL	Value (in engineering units) resulting from the scaling calculation and below which the value entered into the MAInpuGP_CFG.CLOPV input/output needs to be used as the measurement value (PVx outputs). The value that this variable can assume is limited by the low range configured in engineering units (MAInpuGP_CFG.CLOPV). If the range entered from the monitoring system is incorrect (MAInpuGP_CFG.CLOPV = MAInpuGP_CFG.CHIPV = 0.0), it is initialized with the value of the LOPV input.
PVW	WORD	Status word for the various PVx input channels.
LSel	INT	Read/write access Enables to select the selection criterion of the operator (Owner = 1) from the monitoring subsystem. As long as the setpoint is defined by the Program (Owner = 0), the DFB will update the value continuously.
MaxDev	REAL	
Override-Value	REAL	

MAInpuGP_ST.STW Word Structure

Bit	Name	Description
0	Selec-tedChFai-lure	Selected channel detected failure.
1	ChFailure	One or more non-operational channel.
2	DevAlm	Standard deviation is > then maximum deviation Refer to the DevAlm output pin (see EcoStruxure™ Process Expert , Process Control Services User Guide).
3	MaxDevCon-nected	Maximum deviation pin is connected.
15	Rem	Setpoint mode program owner (0=Local, 1=remote).

MAInpuGP_ST.CFGW Word Structure

Bit	Name	Description
0	Override	Read/write access Override mode active
1	Owner	Read/write access Enables to configure whether the setpoint is set by the Program (0) or Operator (1).
14	Maint	Read/write access Maintenance mode active

MAInpuGP_CFG.PVW Word Structure

Bit	Name	Description
0	PV1En	PV1 selection enabled Enabled when a variable or value is connected to input ChIn1.
1	PV2En	PV2 selection enabled Enabled when a variable or value is connected to input ChIn2.
2	PV3En	PV3 selection enabled Enabled when a variable or value is connected to input ChIn3.
3	PV4En	PV4 selection enabled Enabled when a variable or value is connected to input ChIn4.
4	PV1Fail	Detected error on signal PV1. PV1EN = 1 and ChFailure1 = 1.
5	PV2Fail	Detected error on signal PV2. PV2EN = 1 and ChFailure2 = 1.
6	PV3Fail	Detected error on signal PV3. PV3EN = 1 and ChFailure3 = 1.
7	PV4Fail	Detected error on signal PV4. PV4EN = 1 and ChFailure4 = 1.

Public Variables

Public Variable Description

Variable	Type	Description
SC	MAInpuGP_SC_DDT	Provides the frequently needed data to monitor the status of the DFB from the sequential control.

MAInpuGP_SC_DDT Type

Name	Type	Description
PV	REAL	Read-only access Refer to the PV output pin (see EcoStruxure™ Process Expert , Process Control Services User Guide).
LSel	INT	Read/write access. Enables to assign the local setpoint normally for the sequential control if the owner is the Program (Owner input/output is set to 0), and the selected setpoint is Local (SC.Rem public variable is set to 0). Otherwise, the current setpoint (SELD output) is continuously copied to this variable.
IsGood	BOOL	Read-only access

Name	Type	Description
		Refer to the `IsGood` output pin (see EcoStruxure™ Process Expert , Process Control Services User Guide).
`Rem`	`BOOL`	Read/write access Allows the DFB to be configured for a remote setpoint -`RSel`- (1) or local setpoint -`LSel`- (0).
`Owner`	`BOOL`	Read/write access Refer to the `MAInpuGP_ST.CGFW.Owner` input/output pin (see EcoStruxure™ Process Expert , Process Control Services User Guide).

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the management of multi analog input.

Genies

Genie Properties

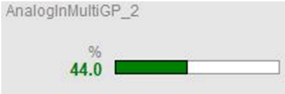
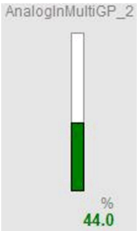
Refer to Genie Properties, page 62.

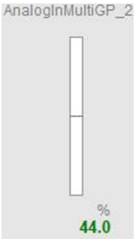
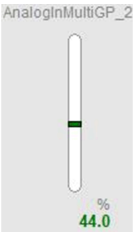
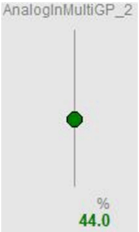
Genie Icons

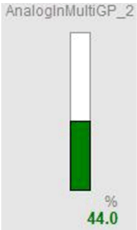
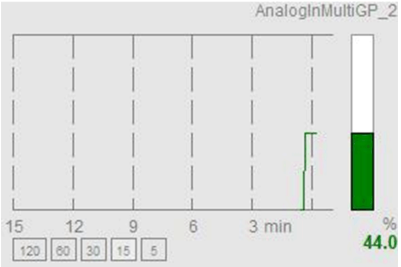
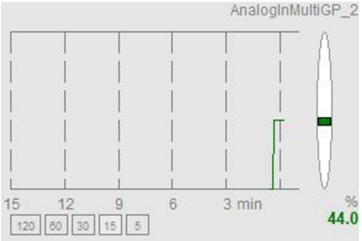
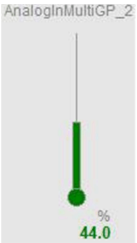
Refer to Genie Icons, page 63.

Representation

The table describes the genies of the `gpl_meters` library:

Graphic symbol	Genie name	Description
	*Bar_PVSP_Horiz*	Displays the label on a horizontal bar with engineering units, present value (PV), and setpoint (SP).
	*Bar_PVSP_Vert*	Displays the label on a vertical bar with engineering units, present value (PV), and setpoint (SP).

Graphic symbol	Genie name	Description
	Bar_PVSP_Trend_Vert	Displays the label on a vertical bar with engineering units, present value (PV), and setpoint (SP). In addition, the symbol displays trends and allows you to select the trend period in minutes.
	*Deviation_PVSP*	Displays the deviation on a vertical bar with engineering units, present value (PV), and setpoint (SP).
	Indicator_PVSP	Displays the label, engineering units, present value (PV), and setpoint (SP).
	*Analyzer_PVSP*	Displays the analyzer on a vertical bar with engineering units, present value (PV), and setpoint (SP).
	Analyzer_PVSP_Trend	Displays the analyzer on a vertical bar with engineering units, present value (PV), and setpoint (SP). In addition, the symbol displays trends and allows you to select the trend period in minutes.
	*Flow_PVSP*	Displays the flow on a vertical bar with engineering units, present value (PV), and setpoint (SP).
	Flow_PVSP_Trend	Displays the flow on a vertical bar with engineering units, present value (PV), and setpoint (SP). In addition, the symbol displays trends and allows you to select the trend period in minutes.

Graphic symbol	Genie name	Description
	*Level_PVSP*	Displays the level on a vertical bar with engineering units, present value (PV), and setpoint (SP).
	*Level_PVSP_Trend*	Displays the level on a vertical bar with engineering units, present value (PV), and setpoint (SP). In addition, the symbol displays trends and allows you to select the trend period in minutes.
	Pressure_PVSP	Displays the pressure on a vertical bar with engineering units, present value (PV), and setpoint (SP).
	*Pressure_PVSP_Trend*	Displays the pressure on a vertical bar with engineering units, present value (PV), and setpoint (SP). In addition, the symbol displays trends and allows you to select the trend period in minutes.
	*Temperature_PVSP*	Displays the temperature on a vertical bar with engineering units, present value (PV), and setpoint (SP).
	Temperature_PVSP_Trend	Displays the temperature on a vertical bar with engineering units, present value (PV), and setpoint (SP). In addition, the symbol displays trends and allows you to select the trend period in minutes.

NOTE: Setpoint is optional parameter and it is hidden, if setpoint item is not available.

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

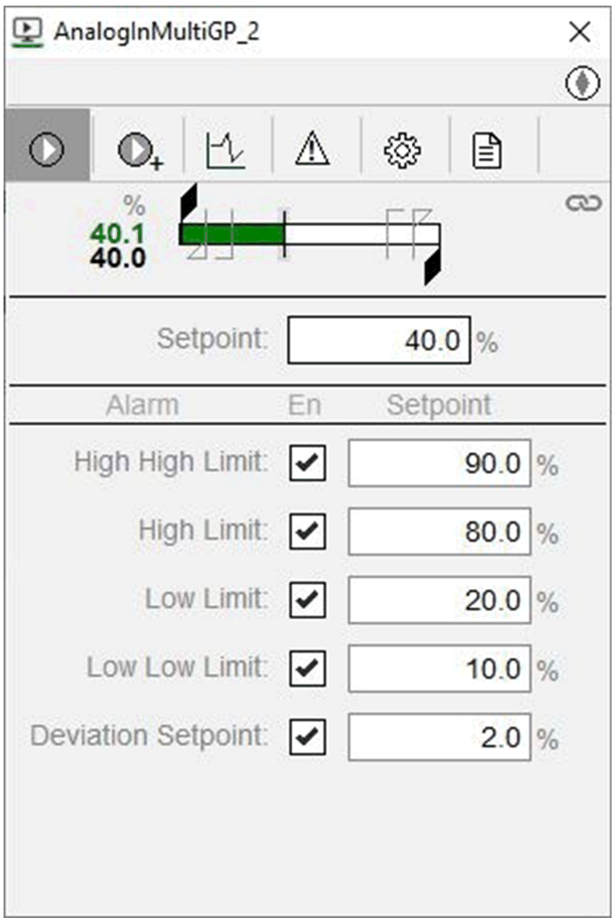
Available Tabs

During operation, clicking an multi analog input genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - ExtOperation
  - Engineering
  - Alarm, page 67
  - Event, page 69
  - Trend, page 69
- Tabs for optional functions, which appear only if configured:
  - Measures, page 70

Operator Tab

The figure shows an operator tab:



Following actions can be performed from an operator tab:

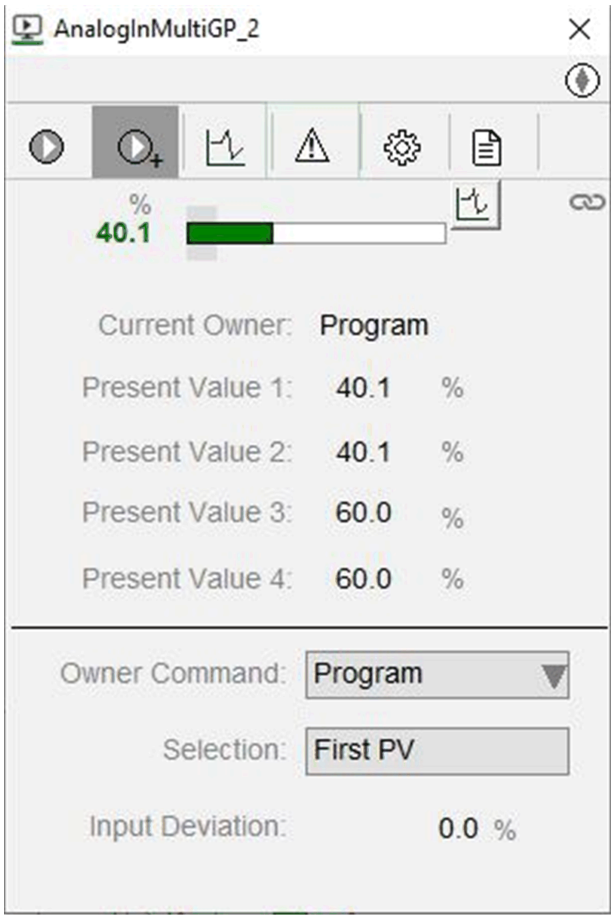
- **En**: Allows you to enable or disable the evaluation of an alarm conditions at the controller level.
- **Setpoint**: Allows you to set the setpoint.

The table describes the control function of an operator tab:

Lable	Item name	Description	Security parameter, page 463
Setpoint	SP	Setpoint	Set setpoint
High High Limit	HighHighLimit, HighHighEnable	Very-high-level setpoint	
High Limit	HighLimit, HighEnable	High-level setpoint	
Low Limit	LowLimit, LowEnable	Low-level setpoint	
Low Low Limit	LowLowLimit, Add LowLowEnable Item	Very-low-level setpoint	
Deviation SP	SPDeviationLimit, HighDeviationEnable, LowDeviationEnable	Deviation limit setpoint	Set setpoint

ExtOperation Tab

The figure shows an extoperation tab:



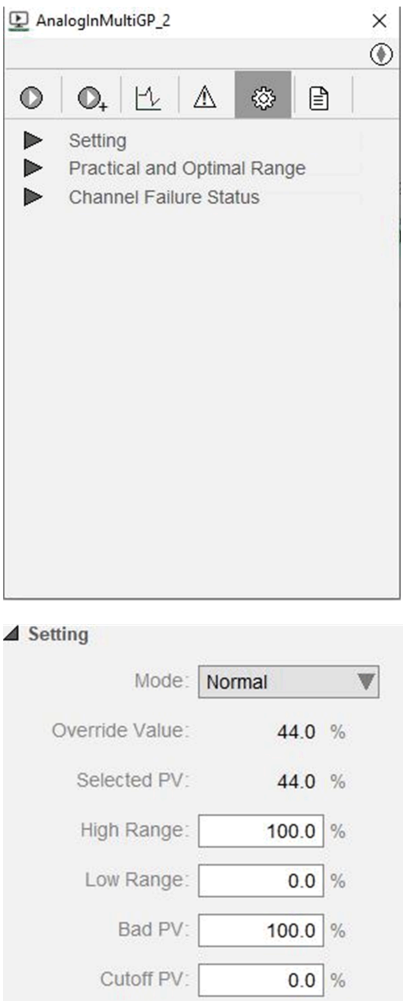
The table describes the control function of an ExtOperator tab:

Lable	Item name	Description	Security parameter, page 463
Current Owner	Owner	Current owner of multi analog input, For more details, refer to operating owner, page 58.	–

Lable	Item name	Description	Security parameter, page 463
Present value1...4	PV	Present value1...4	
Owner Command	OwnerCommand	Owner command of multi analog input (Program, Operator).	
Selection	Selection	Operation for selected analog inputs (Average, Median, Maximum, Minimum).	
Input Deviation	InputDeviation	Maximum deviation allowed.	

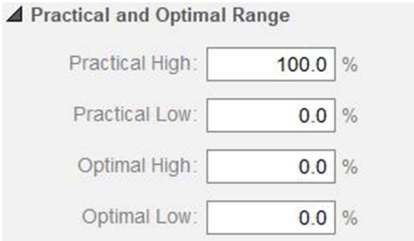
Engineering Tab

The figure shows an engineering tab:



The below label appears on the faceplate when optimal range limits is set to setpoint deviation (2) and practical range disable is unselected.

The below label appears on the faceplate when optimal range limits is set to fixed (1) value and practical range disable is unselected.



The table describes the control function of an engineering tab:

Legend	Item name	Description	Security parameter, page 463
Setting			
Mode	Mode	Allows you to view and change the mode. Refer to Operating Mode, page 59.	Set mode - override, set mode - maintenance.
Override Value	OverrideValue	If override is selected, then input value is equal to external PV, else	Set mode - override

Legend	Item name	Description	Security parameter, page 463
		input value is equal to channel value.	
High Range	PVHigh	Configurable high range.	Set parameter
Low Range	PVLow	Configurable low range.	Set parameter
Bad PV	PVBad	Value to consider when channel inoperable.	Set parameter
Cutoff PV	PVCutOff	Minimum reliable measure.	Set setpoint
Practical and Optimal Range			
Practical High	PRHigh	Practical range high	–
Practical Low	PRLow	Practical range low	–
Channel Failure Status			
`SelectedChFail`	ChannelFailure	One or more channel inoperable.	–
`Channel {x} Failure`	PV{x}	Detected error signal on pv{x}.	–
NOTE: {x} represents the range from 1 to 4.			

Items

Overview

This section describes the variables, abnormal, alarm items, events, and trends item of `multiple analog input`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`SelectedChannelFailure`	Selected channel inoperable	–	*Bit0*
`ChannelFailure`	One or more channel inoperable	Unhealthy/Healthy	*Bit1*
`StdDeviationAlarm`	Deviation alarm	–	*Bit2*
`MaxDevConnected`	Maximum deviation connected	–	*Bit3*
`ProgramSetpointMode`	Setpoint source for program owner	Remote/Sequence	*Bit15*
Bit items derived from ConfigurationWord.			
`InputSource`	Input source selection	Operator/Field	*Bit0*
`OwnerSelect`	Owner of control module	Operator/Program	*Bit1*
Maintenance	Maintenance mode	–	*Bit14*
Bit items derived from PVStatusWord			
`PV{x}En`	PV{x} selection enabled	–	*Bit{y}*

Item name	Description	Enumeration	Address
NOTE: {x} represents the range from 1 to 4. {y} represents the range from 0 to 3.			
PV{x}Fail	Detec{x}1	–	*Bit{4}*
NOTE: {x} represents the range from 1 to 4. {y} represents the range from 0 to 3.			
IO Items			
StatusWord	StatusWord	–	*AnalogInMulti_ MAInputGP_ST.STW*
ConfigurationWord	ConfigurationWord	–	*AnalogInMulti_ MAInputGP_ST.CFGW*
PV	Present value	–	*AnalogInMulti_ MAInputGP_ST.PV*
PV1	Present value 1	–	*AnalogInMulti_ MAInputGP_CFG.PV1*
PV2	Present value 2	–	*AnalogInMulti_ MAInputGP_CFG.PV2*
PV3	Present value 3	–	*AnalogInMulti_ MAInputGP_CFG.PV3*
PV4	Present value 4	–	*AnalogInMulti_ MAInputGP_CFG.PV4*
OverrideValue	Override value	–	*AnalogInMulti_ MAInputGP_CFG. OverrideValue*
PVHigh	Configurable high range	–	*AnalogInMulti_ MAInputGP_CFG. CHIPV*
PVLow	Configurable low range	–	*AnalogInMulti_ MAInputGP_CFG. CLOPV*
PVBad	Value to consider when channel inoperable	–	*AnalogInMulti_ MAInputGP_CFG. BadPV*
PVCutOff	Minimum reliable measure	–	*AnalogInMulti_ MAInputGP_CFG. CutOffPV*
PVStatusWord	PV statusword	–	*AnalogInMulti_ MAInputGP_CFG. PVW*
MaxDeviation	Maximum deviation allowed	–	*AnalogInMulti_ MAInputGP_CFG. MaxDev*
Abnormal	Abnormal		*AnalogInMulti_ MAInputGP_Abnormal*
ORDsp	Optimal range display		*Bit7*
ORHigh	Deviation optimal high		*AnalogInMulti_ MAInputGP_ORHigh*
ORLow	Deviation optimal low		*AnalogInMulti_ MAInputGP_ORLow*
PRHigh	Practical range high		*AnalogInMulti_ MAInputGP_PRHigh*
PRLow	Practical range low		*AnalogInMulti_ MAInputGP_PRLow*
DeviationHighSet-point	High Deviation SP		*AnalogInMulti_ MAInputGP_ DeviationHighSetpoint*

Item name	Description	Enumeration	Address
DeviationLowSet-point	Low Deviation SP		*AnalogInMulti_ MAInputGP_ DeviationLowSetpoint*
Calculated Items			
Owner	Current owner of the multiple analog input. Refer to calculated variable, page 58.		GPL_Owner
Mode	Current mode of the multiple analog input. Refer to calculated variable, page 58.		GPL_ ActionModeAnalo-gInput
Abnormal	Shows abnormal condition of the multiple analog input Refer to Abnormal Conditions, page 480.		GPL_Abnormal

Abnormal Conditions

The table describes the abnormal conditions:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
AbnormalOwner	Abnormal owner status appears, if the current owner is not a normal owner which configured at the instance level.	*Bit11*
AbnormalHighHighEnable	High high alarm enable.	*Bit2*
AbnormalHighEnable	High alarm enable.	*Bit3*
AbnormalLowLowEnable	Low low alarm enable.	*Bit5*
AbnormalLowEnable	Low alarm enable.	*Bit6*
AbnormalMode	Abnormal mode	*Bit1*
AbnormalOperatorTab	Abnormal status of operator tab.	*Bit8*
AbnormalEngineeringTab	Abnormal status of engineering tab.	*Bit12*
AbnormalLowDeviatio-nEnable	Low deviation enable.	*Bit7*
AbnormalHighDeviatio-nEnable	High deviation enable.	*Bit4*
AbnormalDeviationEna-ble	Deviation alarm enable.	*Bit0*
AbnormalRateEnable	Rate enable.	*Bit9*
AbnormalExtendedOpera-torTab	Abnormal status of extended operator tab.	*Bit10*

Alarms

The table describes the alarms that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
SelectedChFailure	Selected channel failure alarm	GPL_BitValueCheck (StatusWord, 0)
StdDeviationAlarm	Standard deviation alarm	GPL_BitValueCheck (StatusWord, 2)
Channel{x} Failure	Channel{x} failure alarm	GPL_BitValueCheck (StatusWord,{y})
NOTE: {x} represents from 1 to 4. {y} represents from 4 to 7.		

Optional Services:

- Refer to AAlarm, page 114

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
StateAlarm	All changes in the multiple analog input status and output.
ModeAlarm	All changes in the mode and owner.
ActionAlarm	All actions by the operator.

Trends

The table describes the trend items that are used by Supervision components:

Item name	Expression	Description
PV	MAInGP_ST.PV	Present value
PV1	MAInGP_ST.PV1	Present value 1
PV2	MAInGP_ST.PV2	Present value 2
PV3	MAInGP_ST.PV3	Present value 3
PV4	MAInGP_ST.PV4	Present value 4

Load cell

What's in This Chapter

Template 482
Control 486
Supervision 503

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the load cell.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$LoadCellGP` template.

Description

General

The `$LoadCellGP` is a high performance weighing module controller which is used for vessels and silos weighing, dosing process and dynamic weighing application on Ethernet IP using Explicit unconnected messaging.

Function Description

The main functions of `$LoadCellGP` template are described in the following table:

Function	Description
Control and Monitoring	When owner is program, functional commands (`Tare`, `Cancel Tare` and `Zero Adjustment`) are executed through user sequence template. Also it enables you to monitor the device. You can control the functional commands from the user sequence DFB.
Device status indication	Displays the status of the device.
Remote resetting	Allows resetting of the device.
Local and remote setpoint	Enables to control the template with a set-point that is determined through a set-point selector (local or remote) in program owner. The local set-point is assigned from the sequential control while the remote set-point is assigned to the template from the logic implemented in the continuous/trigger based control.

Function	Description
Owner	Manages the control system which is the owner (Operator or Program).
Inoperable device management	`Fail` output pin will be high based on the following conditions. It requires `ResetFail`/`Reseting` when device is inoperable. • Inoperable Device • Communication interruption • Configuration interruption `SignalFailure` output pin will be high based on the following conditions. It requires`ResetFail`/`Reseting` when device is inoperable. • `Fail` • OOS mode

Parameters

Parameters

The `$LoadCellGP` template provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Auto reset	`Boolean`	False	• False = Manual reset • True = Auto reset
	Communication detected failure retries	`Integer`	3	Number of retries in case of a communication interruption.
NOTE: Values of boolean parameters are set by using check boxes: • Selected = True • Cleared = False				
Range	High (engineering)	`Float`	10000.0	High range value in engineering units.
	Low (engineering)	`Float`	0.0	Low range value in engineering units.
	Engineering unit	`Enum`	Kg (1)	Engineering units Possible values are g, kg, t, oz,lb, N
	Format	`String`	#####.#	Display format
Time	Device execute order time	`Duration`	00:00:05	Time the device takes to execute commands.
	Maintain `warning` signal		00:00:03	The time duration for which the `Warning` pin signal remains 1.
	Between two auto reset time		00:01:00	The maximum time between two automatic resets.
	Refresh cycle data		00:00:0-05	Time the device takes to refresh the cyclic data.
	Device response time		00:00:05	Time for which client will wait for a response, once the request is initiated, before moving to `Fail` state.
	Inactivity time after retries completion		00:00:10	The period for which the client will not initiate a request after the retries are expired
Setpoint	Force remote setpoint	`Boolean`	False	Enables the remote control operation.
Alarms	For parameters of LimitAlarm, page 118 and DeviationAlarm, page 124.			

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Operation	Normal owner	Enum	Program & Cascade (5)	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	Set owner	Enum	Operator (1)	Privilege to change owner. This parameter is used in Operator Tab, page 505.
	Set setpoint		Operator (1)	Privilege to change setpoint. This parameter is used in Operator Tab, page 505.
	Set Reset		Operator (Confirmed) (10)	Privilege to reset. This parameter is used in Operator Tab, page 505.
	Set parameters		Engineer (5)	Privilege to set parameters. This parameter is used in Operator Tab, page 505.
	Set alarm limits		Operator (1)	Privilege to set alarm limits. This parameter is used in Engineering Tab, page 506.
	Set alarm acknowledge		Operator (1)	Privilege to acknowledge alarm. This parameter is used in Alarm Tab, page 67.
Asset and Display	For more details, refer to the topic describing the parameters of asset and display, page 43.			
Trend	For more details, refer to the topic describing the parameters of trend, page 47			

Data

The table describes the data parameters:

Section	Parameter description	Type	Default	Additional remarks
Range	Optimal range	Enum	Disable (0)	It disables the optimal range indication.
	Practical range disable	Boolean	True	It disables the practical range indication.
Alarm	For more details, refer to the topic describing the parameters of alarm, page 45			
Deadband	Weight	Integer	0	Deadband value for grossPV, Tare PV, and NetPV.
Events	Enable log	Boolean	False	It allows to historize the operator actions and calibration steps.
Trend	Disable gross weight	Boolean	False	It disables gross weight trend tag.
	Disable gross setpoint			It disables gross setpoint trend tag.
Historize	Weight	Boolean	False	It allows to historize the change in weight.
	Setpoint			It allows to historize the change in setpoint.
	Device status			It allows to historize the change in device status, measurement status and response status.

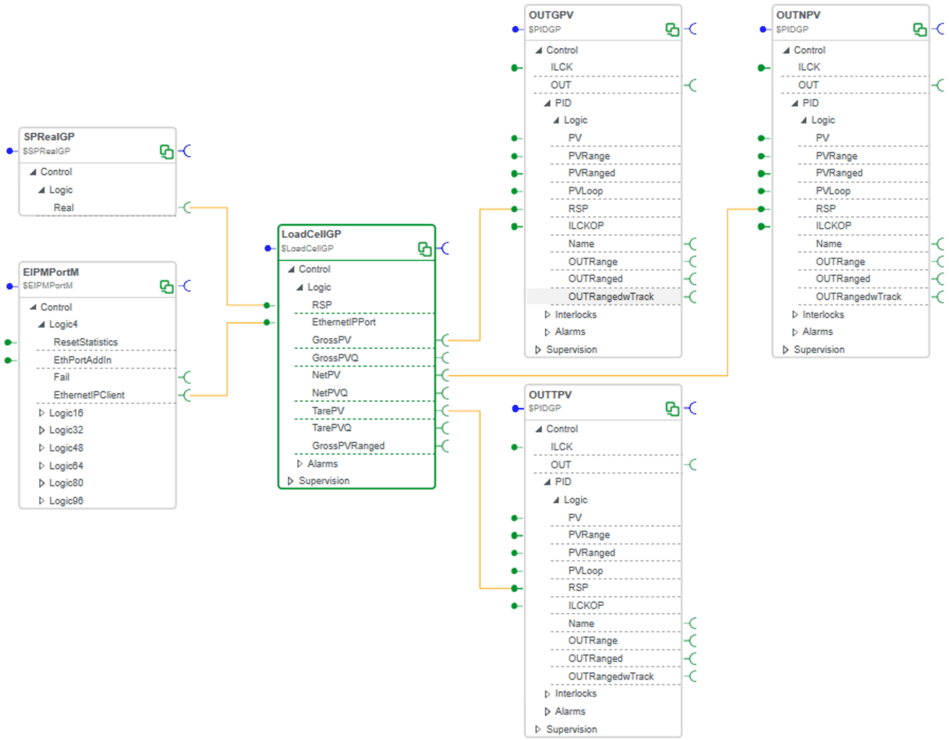
Section	Parameter description	Type	Default	Additional remarks
	`Fail` code			It allows to historize the change in `fail` code.
	`Warning` code			It allows to historize the change in `warning` code.
Alarms	For parameters of LimitAlarm, page 118 and DeviationAlarm, page 124.			

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$LoadCellGP` (also applicable for `$LoadCellCE`) template as it appears in Links Editor and an example of other templates connected to it:



The following link interface create the link symbol on faceplate:

- RSP
- GrossPV, GrossPVQ, NetPV, NetPVQ, TarePV, TarePVQ, and GrossPVRanged

The table describes the control module template interfaces:

Interface identifier	Type/Role	Description	Example
`EthernetIPPort`	*$Real/Ref*	Memory area to communicate with EIP Logical port	–
`GrossPV`	*$Real/Def*	Gross Weight	–
`GrossPVQ`	*$PVQGP/DO*	Gross Weight with quality	–
NetPV	*$Real/Def*	Net Weight	–
`NetPVQ`	*$PVQGP/DO*	Net Weight with quality	–

Interface identifier	Type/Role	Description	Example
TarePV	*$Real/Def*	Tare Weight	–
TarePVQ	*$PVQGP/DO*	Tare Weight with quality	–
GrossPVRanged	*$RangedValue/Def*	Gross weight with range	–

Control

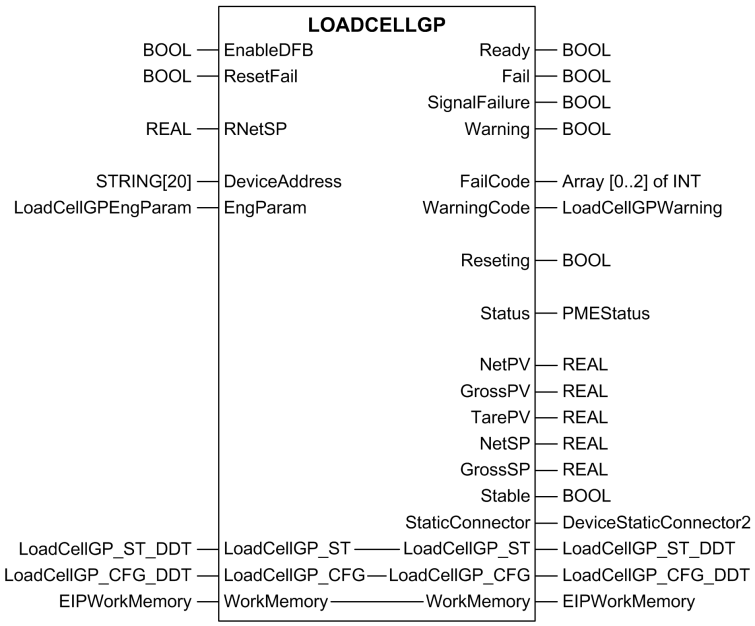
Overview

This chapter describes the `LOADCELLGP` DFB.

DFB Representation

Representation

The following figure represents the functional module of `LOADCELLGP`.



Inputs

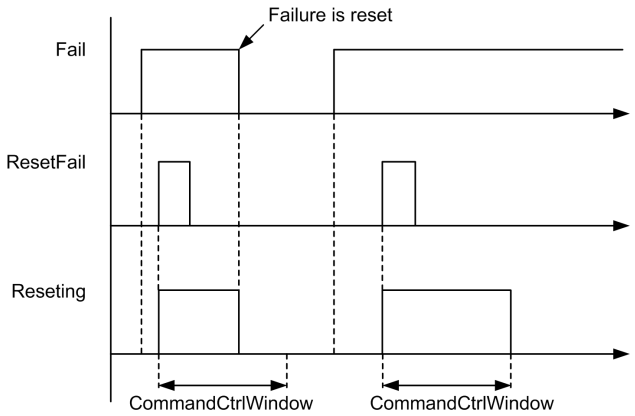
Input Parameter Description

Parameter	Type	Description
`EnableDFB`	`BOOL`	This input enables the normal execution of the control block. 0 = The entire DFB is restarted (statuses and output values are lost) and output values are set to 0 and OOS mode is set to true. 1 = Enables communication with the devices for their operation.
`ResetFail`	`BOOL`	1 = Resets the `Fail` and `SignalFailure` output pin to 0. **NOTE:** • When there is a detected configuration mismatch, output pins `Fail` and `SignalFailure` cannot be reset to 0.

Parameter	Type	Description	
		• When Owner is in Program and no detected failure, output pins `Fail` and `SignalFailure` can be reset to 0.	
`RNetSP`	`REAL`	Remote setpoint	
`DeviceAddress`	`STRING[20]`	Device address on Ethernet IP network	
`EngParam`	`LoadCellGPEngParam`	Engineering parameters.	
	`LoadCellGPParam`	The structure holds data containing the engineering parameter that the block requires. (constants enabled). The following table describes parameter information.	
	Parameter	**Type**	**Description**
	`CommandCtrlWindow`	`TIME`	Control time for operations. This is the time that the block waits for the operations to be carried out by the device. If a command has been sent and the command is not executed within the time indicated by this variable, an alarm is issued. The commands that are controlled are `Tare`, `Cancel Tare`, and `Zero Adjustment`. In the event of a ResetFail, this is not interpreted as an alarm. Instead, the detected failure continues, and you have to reset the `Reseting` output.
`EngParam`	`Refresh`	`TIME`	Refresh time for device data on Ethernet IP communications. **NOTE:** This refresh operation is carried out on read variables. Write requests are carried out when needed.
	`Scantime`	`TIME`	Allows you to configure the time for which the alarm signals are kept active. Helps the monitoring subsystem to acquire the data for the alarms that are automatically reset.
	`MaxResetTime`	`TIME`	When in automatic `ResetMode`, this variable is used to define the maximum time that can elapse between 2 consecutive resets.
	`EngUnit`	`UINT`	User defined Engineering unit. Possible values are: • 0 = g • 1 = kg • 2 = t • 3 = oz • 4 = lb • 5 = N **NOTE:** • DTM calibration window allows you to configure **µdef** and **N.m** unit. This measurement unit is mainly used in strain gauge and torque based applications respectively. Since this control function will be used for weighing application. **µdef** and **N.m** unit are not taken into consideration. • If user enters other than predefined engineering unit, then control function will consider the last configured engineering unit.
`EngParam`	`ResetMode`	`BOOL`	Enables to configure the type of reset. This type of reset is used for communication interruption and device interruption. The time defined in `CommandCtrlWindow` is used to define the interval after which a reset has to be carried out. The first reset is carried out after the time defined in `CommandCtrlWindow` elapses. The second reset is carried out after `CommandCtrlWindow` * 2 elapses, so on. If the value of `CommandCtrlWindow` is 0 s, its value is not used and is instead replaced with a value of 1 s. The following table describes the type of the reset:
		Value	**Description**
		False	Communications with the device is reset with the `ResetFail` variable.
		True	Communications with the device is reset automatically.
	`EIPClientEngParam`	The structure holds data containing the engineering parameter that the client block requires. The following table describes parameter information.	
	Parameter	**Type**	**Description**

Parameter	Type	Description	
	CommFailRetries	INT	Number of retries in case of a communication interruption.
	Timeout	TIME	Time for which client will wait for a response, once the request is initiated, before moving fail state.
	InactivityTime	TIME	The period for which the client will not initiate a request after the retries are expired.

Timing Diagram



Outputs

Output Parameter Description

Parameter	Type	Description	
Ready	BOOL	1 = The device is enabled and free of detected errors. The device is ready to carry out weighing functional commands (Tare, Cancel Tare and Zero Adjustment).	
Fail	BOOL	1 = A detected failure in the control block or in the device or a communication interruption. To reset the Fail output pin, the ResetFail input has to be activated. The last detected error code is shown on FailCode. **NOTE:** If a communication interruption occurs, the variables being read from the device cease to be refreshed as a refresh operation can no longer be carried out. The variables keep their last value.	
SignalFailure	BOOL	1 = A detected failure in the control block or in the device or a communication interruption or OOS mode is enabled. **NOTE:** To reset the Fail output pin, the ResetFail input has to be activated. The last detected error code is shown on FailCode.	
Warning	BOOL	1 = An alarm has been activated for the device. It cannot be reset because the signal remains active until the cause of the alarm is removed or until the maximum set ScanTime is reached.	
FailCode	Array [0..2] of INT	When the Fail output is 1, it holds the code for the detected error. If Fail bit is 0, it indicates the last detected error that occurred. The detected error source is specified by using a 3 level structure. Refer to Diagnostic Information Management (see EcoStruxure™ Process Expert, Device Control Services User Guide) for additional information.	
WarningCode	PMESWTWarning	Holds a data structure with the alarm information currently on the device. The following table describes the WarningCode	
	Parameter	**Type**	**Description**
	Device	BOOL	Alert when executing weighing functional commands (Tare, Cancel Tare and Zero Adjustment).
	Order	BOOL	1 = An alarm. The device is not responding to the command within the time specified in CommandCtrlWindow.
Reseting	BOOL	1 = A reset is being carried out.	

Parameter	Type	Description	
		The `CommandCtrlWindow` variable indicates the maximum time for resetting the detected failure. When a communication reset is carried out with `ResetFail`, the DFB tries to reset the detected failure within the time period defined in `CommandCtrlWindow`. If the detected failure is reset, the `Fail` and `Reseting` output variables are reset (set to `FALSE`). If the detected failure is not reset, the `Reseting` variable is set to `FALSE` and the `Fail` variable remains `TRUE`. The `ResetFail` is edge-based.	
`Status`	`LoadCellGPStatus`	The structure holds data containing the information that the function block extracts from the status variable of the weighing module.	
	Parameter	**Type**	**Description**
	`DeviceStatus`	`WORD`	Information on device calibration of the weighing module.
	`MeasurementStatus`	`WORD`	Measurement and device status of the weighing module.
	`ResponseStatus`	`WORD`	Response status provides the state of the command currently being processed by the device.
`NetPV`	`REAL`	Net measurement value in engineering units.	
`GrossPV`	`REAL`	Gross measurement value in engineering units.	
`TarePV`	`REAL`	Tare measurement value in engineering units.	
`NetSP`	`REAL`	Net setpoint value in engineering units.	
`GrossSP`	`REAL`	Gross setpoint value in engineering units. GrossSP =TarePV + NetSP	
`Stable`	`BOOL`	Grosss PV is stable and it is not equal to TarePV.	
`StatisticConnector`	`DeviceStatistic-Connector2`	`StatisticConnector` has two client data, Read client and Write client.	
	Parameter	**Type**	**Description**
	`ReadClient`	`Statistic-Connector1`	`StatisticConnector` of Read client.
	`WriteClient`	`Statistic-Connector1`	`StatisticConnector` of Write client.
	`StatisticConnec-tor1`	Information data is used with EIP communication to obtain statistics on the bus network (requests carried out and time between requests). This structure has been created for its use together with the `StatisticCounter` DFB in General Purpose library for communication.	
	Parameter	**Type**	**Description**
	`Start`	`BOOL`	1 = The operation has started.
	`EndOk`	`BOOL`	1 = The operation has ended successfully.
	`EndNOk`	`BOOL`	1 = The operation has ended with a detected error.
	`TotalTime`	`INT`	Total time taken for current request.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
LoadCellGP_ST	LoadCellGP_ST_DDT	Device data structure that holds the minimum information required for performing control and monitoring functions. The information used by the operator screen is readable/writable from the HMI or SCADA system.
LoadCellGP_CFG	LoadCellGP_CFG_DDT	Data structure with the device information. The information is used by the operator screen and is readable from the HMI or SCADA system.
Workmemory	EIPWorkMemory	Memory area which contains the client reference is used for Ethernet/IP communications.

LoadCellGP_ST_DDT

Parameter	Type	Description
NetPV	REAL	Net measurement value in engineering units.
GrossPV	REAL	Gross measurement value in engineering units.
TarePV	REAL	Tare measurement value in engineering units.
NetSP	REAL	Net Setpoint value in engineering units.
GrossSP	REAL	Gross Setpoint value in engineering units.
OptimalDev-HighSP	REAL	Optimal deviation high setpoint.
OptimalDevLowSP	REAL	Optimal deviation Low setpoint.
STW	WORD	Provides the device status. Access to the data held in this bit word is read-only.
CFGW	WORD	Device control. Provides the necessary means to control the device from the supervision in operator mode (Owner = 1).
EngUnit	UINT	Unit for Processing the measurement values in Engineering unit. Possible values are: • 0 = g • 1 = kg • 2 = t • 3 = oz • 4 = lb • 5 = N

LoadCellGP_CFG_DDT

Parameter	Type	Description
MaxCapacity	UDINT	Maximum capacity of the weighing module.
CalibratedMaxCa-pacity	UDINT	Last calibrated maximum capacity of the weighing module.
CalibrationLoad	UDINT	Calibration load for calibration the device.
NetSP	REAL	Net setpoint value in engineering unit.
OptimalDevSPHigh	REAL	Optimal deviation high setpoint value.
OptimalDevSPLow	REAL	Optimal deviation low setpoint value.

Parameter	Type	Description
ViewSelection	BOOL	0 = Gross view, 1 = Net view Selection can be done when Owner = 0.
MeasurementStatus	WORD	Measurement and device status of the weighing module.
DeviceStatus	WORD	Informations on Device calibration of the weighing module.
ResponseStatus	WORD	Response status provides the state of the command currently being processed by the device.
Warningcode	WORD	Alarm code information. Captures the values from the WarningCode output.
Failcode0	INT	Code of last level 0 detected error. Indicates that a detected error has occurred FailCode[0].
Failcode1	INT	Code of last level 1 detected error. Indicates that a detected error has occurred FailCode[1].
Failcode2	INT	Code of last level 2 detected error. Indicates that a detected error has occurred FailCode[2].
ScaleInterval	UINT	Minimal difference between two consecutive indicated values (either gross or net). Read & write access. Possible value are: 1, 2, 5, 10, 20, 50
CalibratedScaleInterval	UINT	Indicates last calibrated Scale interval parameter.
Stepcalibration	UINT	Number of the step calibration in progress. • 1 = Calibration parameter mismatch • 2 = Physical calibration done • 3 = Zero acquisition done. • 4 = Load - 1 acquisition done. • 5 = Calibration done and parameters saved. • 6 = Calibration canceled. • 7 = Device not responding. **NOTE:** In Step Calibration = 6, the last calibrated values are returned back to the device register.
CalibratedCalibrationUnit	UINT	Indicates the lasts calibrated unit.
CalibrationUnit	UINT	Engineering unit which is used for calibrating the device. Read and write access. Possible values are: • 0 = g • 1 = kg • 2 = t • 3 = oz • 4 = lb • 5 = N

LoadCellGP_CFG.MeasurementStatus Structure

Bit	Description
0	Measurement is OK.
1	Gross weight is less than negative Maximum capacity.
2	Gross weight is greater than maximum capacity.
3	Converter saturation or Analog signal out of A/D input range.
4	0 = Measurement is unstable. 1 = Measurement is stable.

Bit	Description
5	Measurement out of ¼ division flag.
6	0 = Memory Ok. 1 = Detected memory `failure`.
7	0 = none. 1 = At least one preset tare is processed.
8	0 = Logical Input I0 state is disabled. 1 = Logical Input I0 state is enabled.
9	0 = Logical Input I1 state is disabled. 1 = Logical Input I1 state is enabled
10	0 = Logical Output S0 state is disabled. 1 = Logical Output S0 state is enabled.
11	0 = Logical Output S1 state is disabled. 1 = Logical Output S1 state is enabled.
12	0 = Logical Output S2 state is disabled. 1 = Logical Output S2 state is enabled.
13	0 = Logical Output S3 state is disabled. 1 = Logical Output S3 state is enabled
14	Tare process flag.
15	Zero/Tare process flag.

`LoadCellGP_CFG.DeviceStatus` Structure

Bit	Description
0	User dosing cycle in progress.
1	At least one user calibration is processed.
2	Physical calibration is set.
3	Theoretical calibration is set.
4	Load cell wiring mode: • 0 = 4 wire • 1 = 6 wire
5	Termination resistor is set.
6	Theoretical zero done.
7	Theoretical calibration done.
8	Zero adjustment done.
9	Span adjustment done.
10	Start physical calibration done.
11	Zero acquisition done.
12	Load 1 acquisition done
13	Load 2 acquisition done.
14	Load 3 acquisition done.
15	Save calibration done.

LoadCellGP_CFG.ResponseStatus Structure

Bit	Description
0	Ready to accept a new command.
1	Command execution in progress.
2	Command execution complete.
3	Detected error during command execution.

LoadCellGP_ST.STW Structure

Bit	Description
0	Unknown state.
1	Not ready.
2	-
3	Inoperable device.
4	Detected `warning`.
5	Communication interruption.
6	Requires `Reseting`.
7	Rem
8	`Reseting`.
9	Enable DFB.
10	`SignalFailure`.
11	Stability.
12	Cancel calibration disable.
13	`Rearm` enable.
14	`ExtCalibration` - If OOS not enabled & calibration parameters are modified. Then this bit is triggered to enable **Accept Device Values**.

LoadCellGP_ST.CFGW Structure

Bit	Description
0	Reset `Fail`.
1	Owner. • 0 - Program • 1 - Operator
2	Tare command.
3	Cancel tare command.
4	Zero adjustment command.
5	Physical calibration command.
6	Validation step calibration.
7	Cancel calibration command.
10	Accept new values.
15	OOS mode. Enables to determine whether the device is out of service (1) or in use (0).

Warning Bit Structure

Bit	Description
0	Refer to `Device` in the `WarningCode (see EcoStruxure™ Process Expert, Device Control Services User Guide)` output pin .
1	Refer to `Order` in the `WarningCode (see EcoStruxure™ Process Expert, Device Control Services User Guide)` output pin.

Public Variables

Public Variable Description

Parameter	Type	Description
`SC`	`LoadCellGP_SC_DDT`	Provides the frequently needed data for monitoring and controlling the DFB from sequence control.

LoadCellGP_SC_DDT

Parameter	Type	Description
`NetPV`	`REAL`	Net measurement value in Engineering unit.
`GrossPV`	`REAL`	Gross measurement value in Engineering unit.
`TarePV`	`REAL`	Tare measurement value in Engineering unit
`NetsP`	`REAL`	Measurement setpoint value in Engineering unit.
`GrossSP`	`REAL`	Gross measurement setpoint value in Engineering unit
`OptimalDevSPHigh`	`REAL`	Optimal deviation high setpoint value.
`OptimalDevSPLow`	`REAL`	Optimal deviation low setpoint value.

Parameter	Type	Description
View-Selection	BOOL	0 = Gross view, 1 = Net view Selection can be done when Owner = 1.
Properties	WORD	Bits are expressed as ranked bit. • Bit 0 (Owner): Provides the data needed to configure the DFB normally from the monitoring subsystem. ◦ Read access 0 = Program ◦ 1 = Operator • Bit 1 (OOS): 1 = Out-Of-Service mode is selected. Read access. • Bit 2 (Fail): A detected failure in the control block or in the device or a communication interruption. • Bit 3 (ConfigFail): Configuration mismatch. • Bit 4 (MeasureStable): 1 = Measurement is stable, 0 = Measurement is not stable. Read only. • Bit 5 (TareDone): 1 = At least one tare is processed. Read only. • Bit 6 (ZeroDone): 1 = At least one zero is processed. Read only. • Bit 7 (CMDReady): Ready to accept a new command. Ready only. • Bit 8 (CMDInProgress): Command execution in progress. Ready only. • Bit 9 (CMDComplete): Command execution complete. Ready only. • Bit 10 (CMDError): Detected error during command execution. Ready only. • Bit 11: (TareCMD): Execution of tare command. Read/ Write access. • Bit 12 (CancelTareCMD): Execution of cancel tare command. Read/ Write access. • Bit 13 (ZeroADJCMD): Execution of Zero adjustment command. Read/ Write access. • Bit 14 (Rem): Read/ Write access. ◦ 1 = Enables the DFB to be configured to remote set-point. ◦ 0 = Enables the DFB to be configured to local set-point. • Bit 15 (Stable): Read only. Grosss PV is stable and it is not equal to TarePV.

Diagnostic Information Management

Overview

The diagnostic codes the device can return are read from the FailCode output variable.

Parameter Configuration Diagnostic Codes

Following are the Parameter configuration diagnostic codes:

- FAILCODE [0]: 16#0003
- FAILCODE [1]: 16#0000
- FAILCODE [2]: 16#0004

The above mentioned parameter configuration diagnostic codes occurs due to the given below conditions:

- If user enters other than predefined engineering units through other source (DTM).

- If there is a difference in calibrated calibration parameters from device register parameters and when Out Of Service mode is not enabled.
- When Out Of Service mode is enabled to start the calibration operation.

EthernetIP Communication Diagnostic Codes

This code indicates that communications have not been established and can be reset:

- `FailCode[0]`: 16#0002
- `FailCode[1]`: 16#0000
- `FailCode[2]`: 16#0004

After the communications have been established, check EthernetIP client diagnostic codes for `FailCode [0]` and `FailCode [1]`. The components make a distinction between detected read request and write request interruptions:

- `FailCode[2]`: 16#0001 Read
- `FailCode[2]`: 16#0002 Write

Device Diagnostic Codes

Following are the Device diagnostic codes:

- `FailCode[0]`: 16#0001
- `FailCode[1]`: 16#0000
- `FailCode[2]`: 16#0005

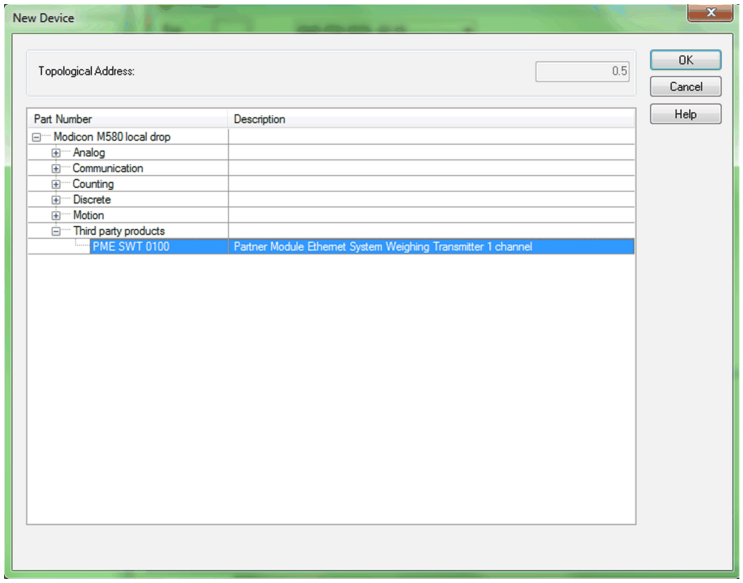
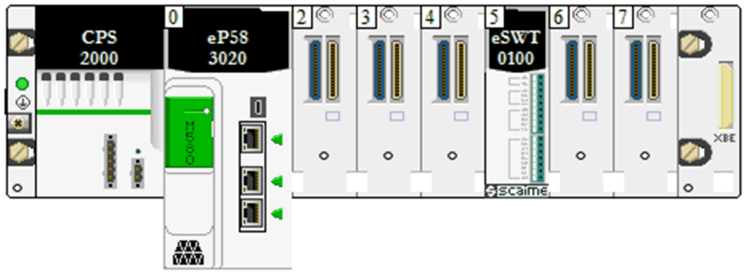
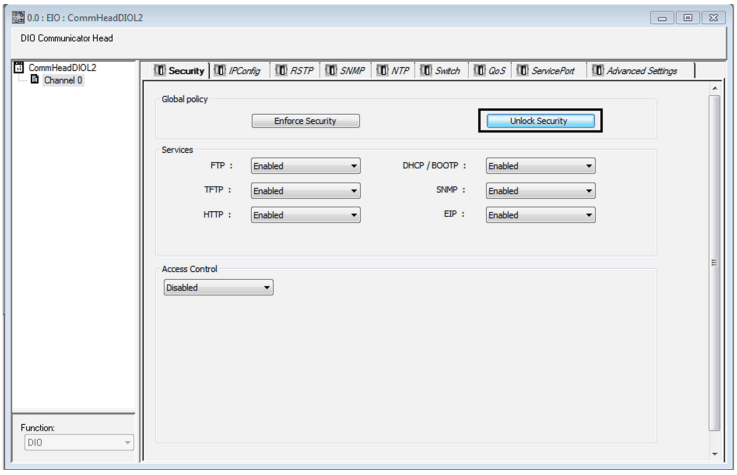
The above mentioned device diagnostic codes occurs due to the given below conditions:

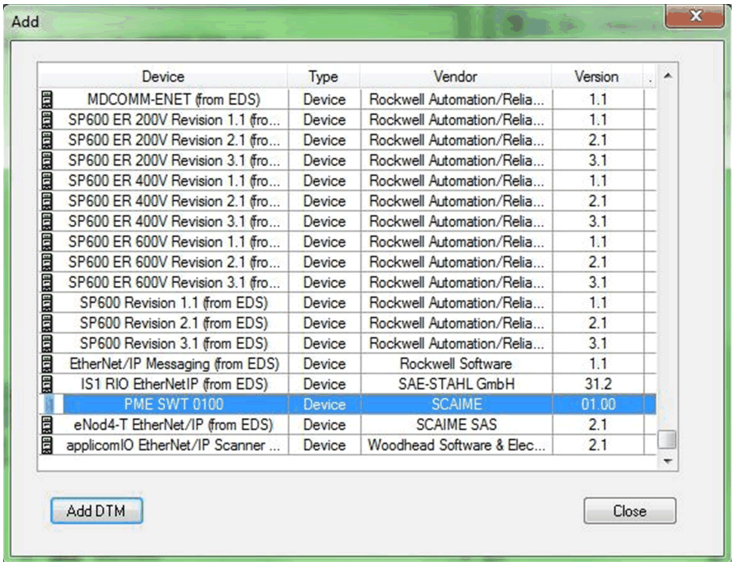
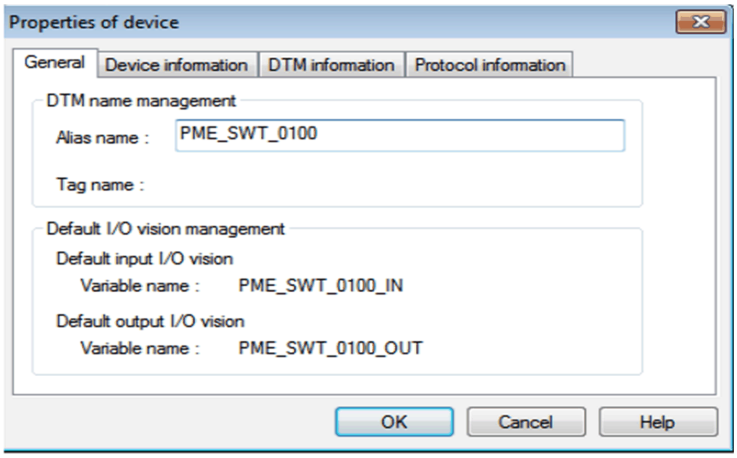
- Detected memory `failure`.
- Gross weight is less than negative maximum capacity.
- Gross weight is greater than maximum capacity.
- Converter saturation or analog signal out of A/D input range.

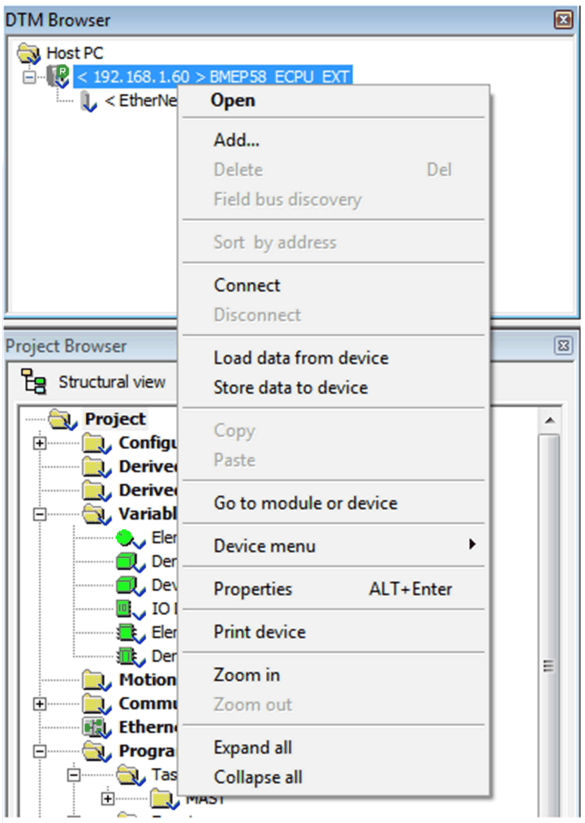
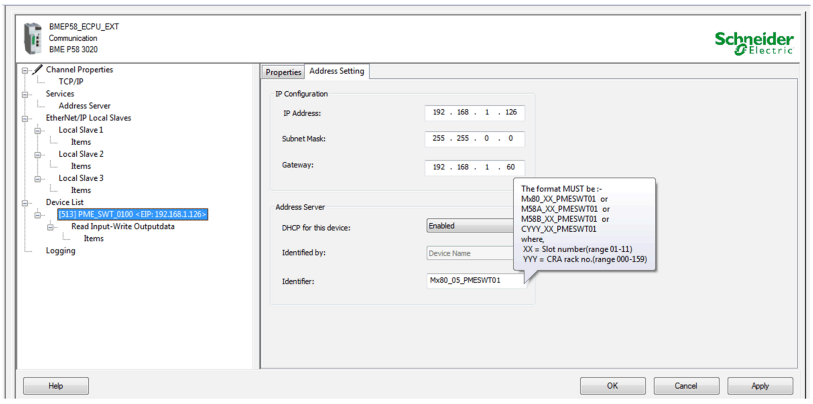
Configuration of PMESWT Module

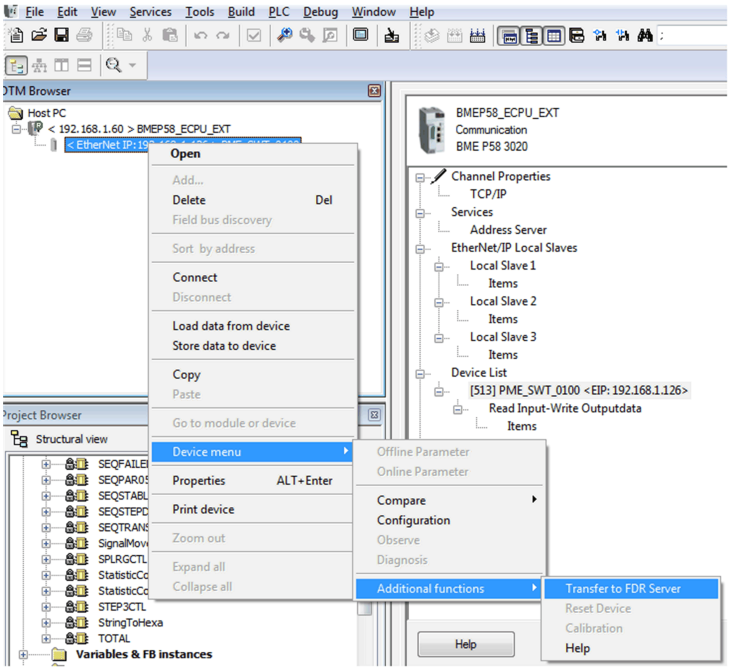
Procedure to Configure PMESWT Module on M580 Controller

Step	Action
1	Create new Control Expert project by selecting M580 CPU.
2	Add PME SWT module by right clicking on empty slot of M580 Ethernet backplane and select PME SWT module from new device window.

Step	Action
	  **NOTE:** Verify that you have not selected slot number 2, 8, 10 and 11 to add PME SWT device as Ethernet Port is absent in those slots.
3	Navigate to PLC bus (**Project Browser** →**Configuration** →**PLC bus**). Click on the **Ethernet Port** available on the CPU which opens the port configuration. On the Security tab, click the **Unlock Security** button to enable **FTP**, **TFTP** and **HTTP** communication. 
4	Open DTM browser screen from **Menu Tools** → **DTM Browser**. Right click the master DTM in the DTM browser and select **PME SWT 0100** device DTM and click **Add DTM**.

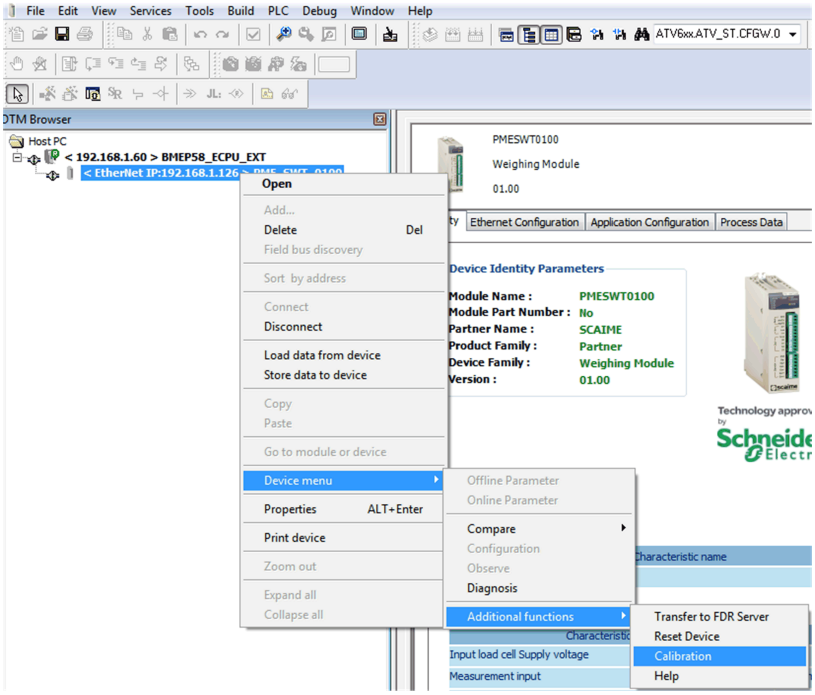
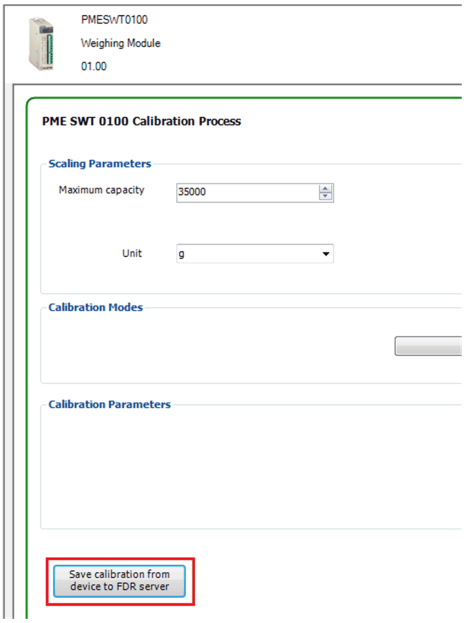
Step	Action
	
5	Provide the **Alias name** for the DTM of your choice and click **OK** on **Properties of device** window. 
6	In the **DTM browser**, right click on the CPU master DTM and click **Open**. The Master DTM properties window opens.

Step	Action
	
7	In the Master DTM properties window, navigate to **Device List** → **PMESWT**, select **Address Setting** tab to configure the device IP address and address server configuration and then click **Apply**. 
8	Build and transfer the Control Expert configuration file to the PLC.
9	In the DTM browser, right click on the M580 CPU DTM and then select **Connect**.
10	In the DTM browser, right click on the PMESWT device DTM and then navigate to **Device menu** → **Additional functions** → **Transfer to FDR Server**.

Step	Action
	
11	After successful transfer of FDR, device will become healthy and the following message window displays.
12	To verify that the device is connected or not, In the DTM browser, right click on the PMESWT device DTM and select **Connect**. Now, the IP address and name of PMESWT module appears in bold lettering, indicating both devices are connected.

Procedure to save Calibration from Device to FDR Server

It is recommended to perform the below steps at the end of each calibration from Supervision.

Step	Action
1	In the DTM browser, right click on the PMESWT device DTM and then navigate to **Device menu** → **Additional functions** → **Calibration**. 
2	In the DTM calibration window, click **Save calibration from device to FDR server**. 

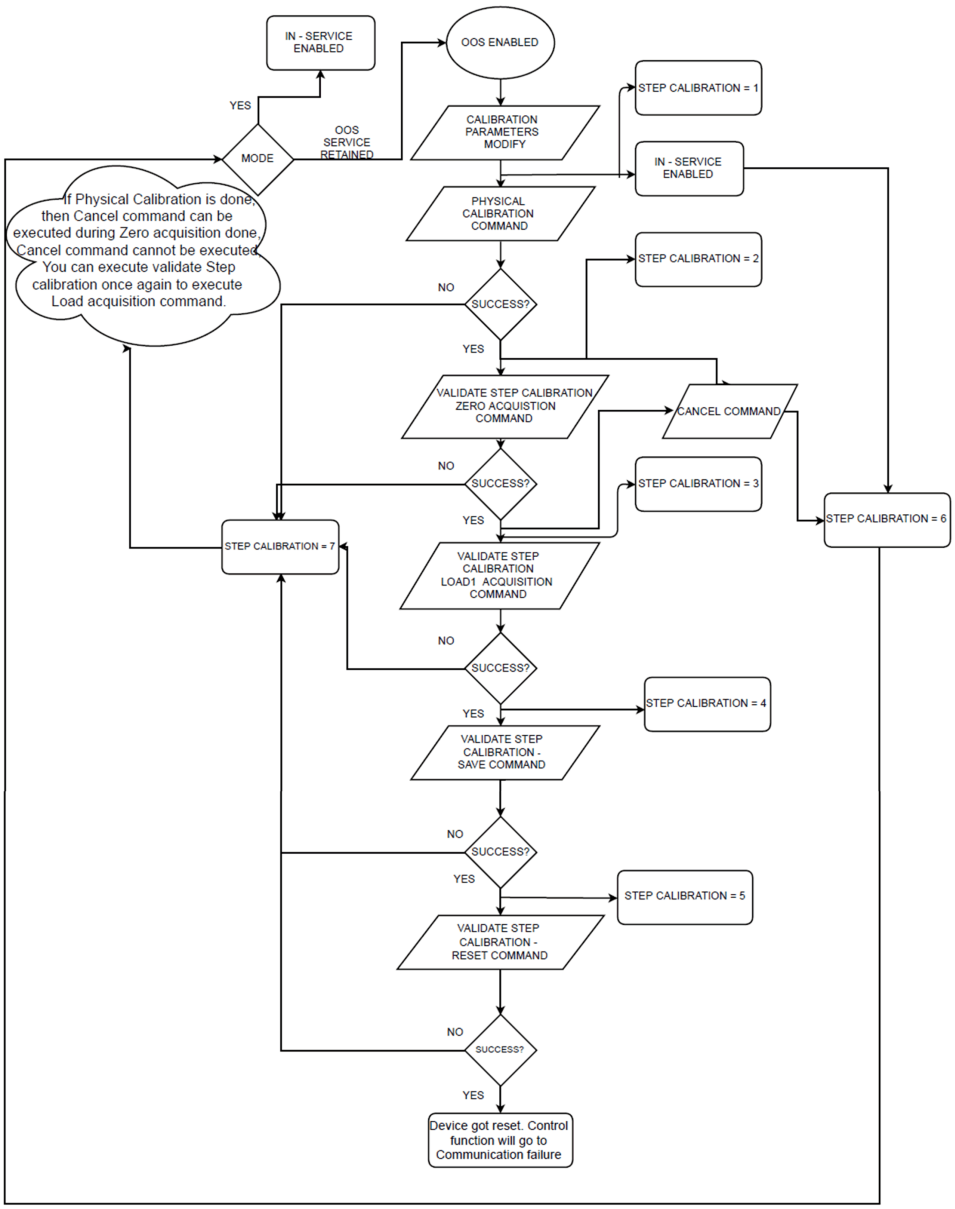
PMESWT DFB Work Flow

Using the Control Function Block

Step	Action
1	During rising edge of `EnableDFB` or first download, control function is set to Owner (Program) and Mode (In-Service).
2	Once the device is ready for operation, control function reads the measurement values and calibration parameters from the device and process the EU conversion accordingly.

Step	Action
3	If you want to perform calibration, then you need to set Owner (Operator) and Mode (Out Of Service). For execution of calibration commands refer Weighing Module Calibration Commands Flow Chart, page 502.
4	For execution of functional commands refer Weighing Module Functional Commands Flow Chart, page 503.
5	During operation, if you perform calibration from external source (DTM), then control function will generate detected configuration failure because the last calibrated value and new value read from device register are different (calibration parameter mismatch). In such scenario, if you have performed calibration through external source (DTM), then **Accept Device Values** button in the calibration faceplate gives flexibility to accept the calibrated values. **Accept Device Values** button is enabled only when there is calibrated parameter mismatch (detected configuration failure). You can clear this detected failure by clicking **Accept Device Values** button or start calibration procedure from the faceplate, by taking the object in Out Of Service mode.

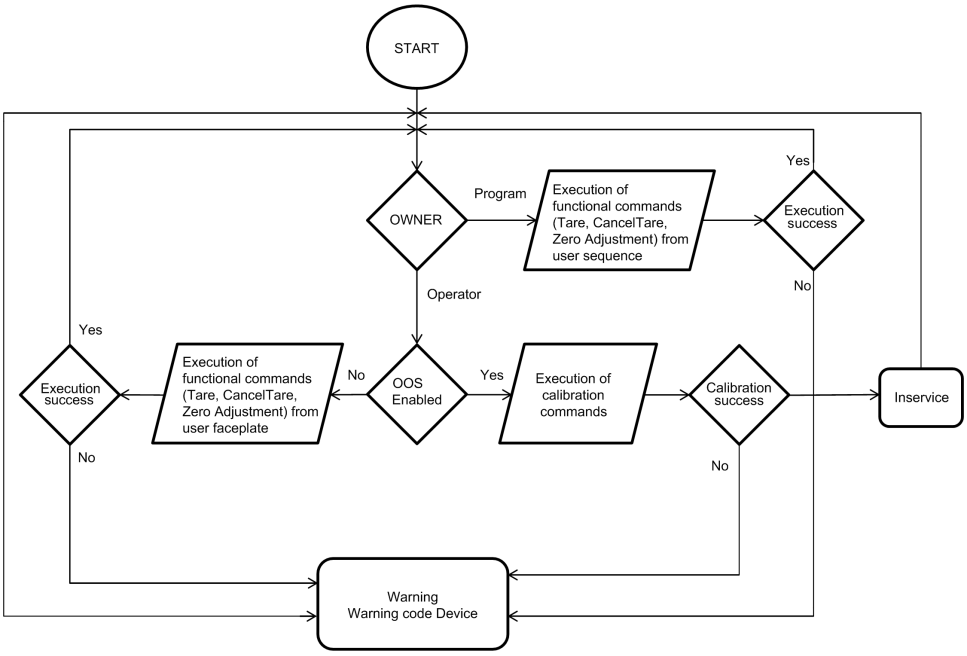
Weighing Module Calibration Commands Flow Chart



NOTE:

- Initially you have to configure the weighing module through **DTM**.
- It is suggested to perform the calibration from the faceplate only.
- Before the start of calibration you need to verify that calibration parameters are entered properly.
- In Load acquisition state (after zero acquisition done) during calibration you need to verify that a physical load is placed on the load cell.
- At the end of each calibration, once the device reset is performed and before giving `ResetFail` you need to verify that the physical device is healthy.
- In Step Calibration = 6, the last calibrated values are returned back to the device register.

Weighing Module Functional Commands Flow Chart



Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the management of loadcell.

Genies

Genie Properties

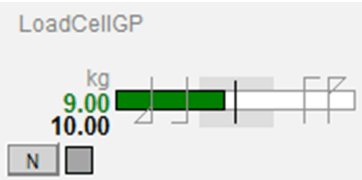
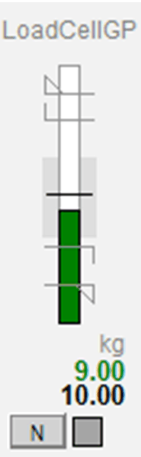
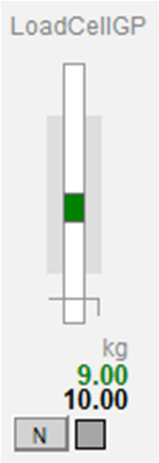
Refer to Genie Properties, page 62.

Genie Icons

Refer to Genie Icons, page 63.

Representation

The table describes the genies of the `gpl_loadcell` library:

Graphic symbol	Genie name	Description
	*Bar_PVSP_Horiz*	displays present value (PV) and setpoint (SP) display on a horizontal bar.
	*Bar_PVSP_Vert*	Displays present value (PV) and setpoint (SP) display on a vertical bar.
	*Deviation_PVSP*	Displays deviation on a vertical bar with present value (PV) and setpoint (SP).

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

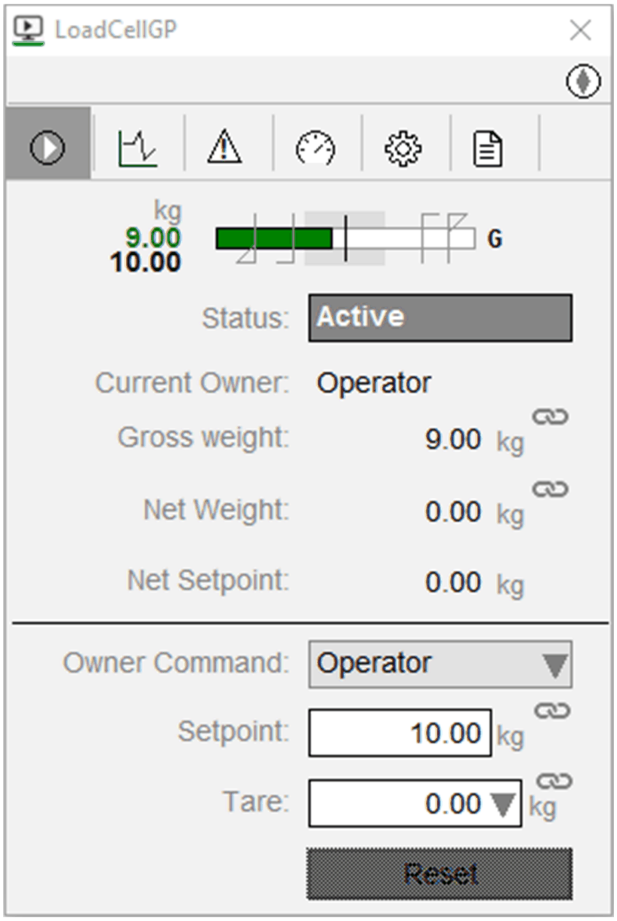
Available Tabs

During operation, clicking a loadcell genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
  - Event, page 69
  - Trend, page 69
  - Measures, page 70

Operator Tab

The figure shows an operator tab:



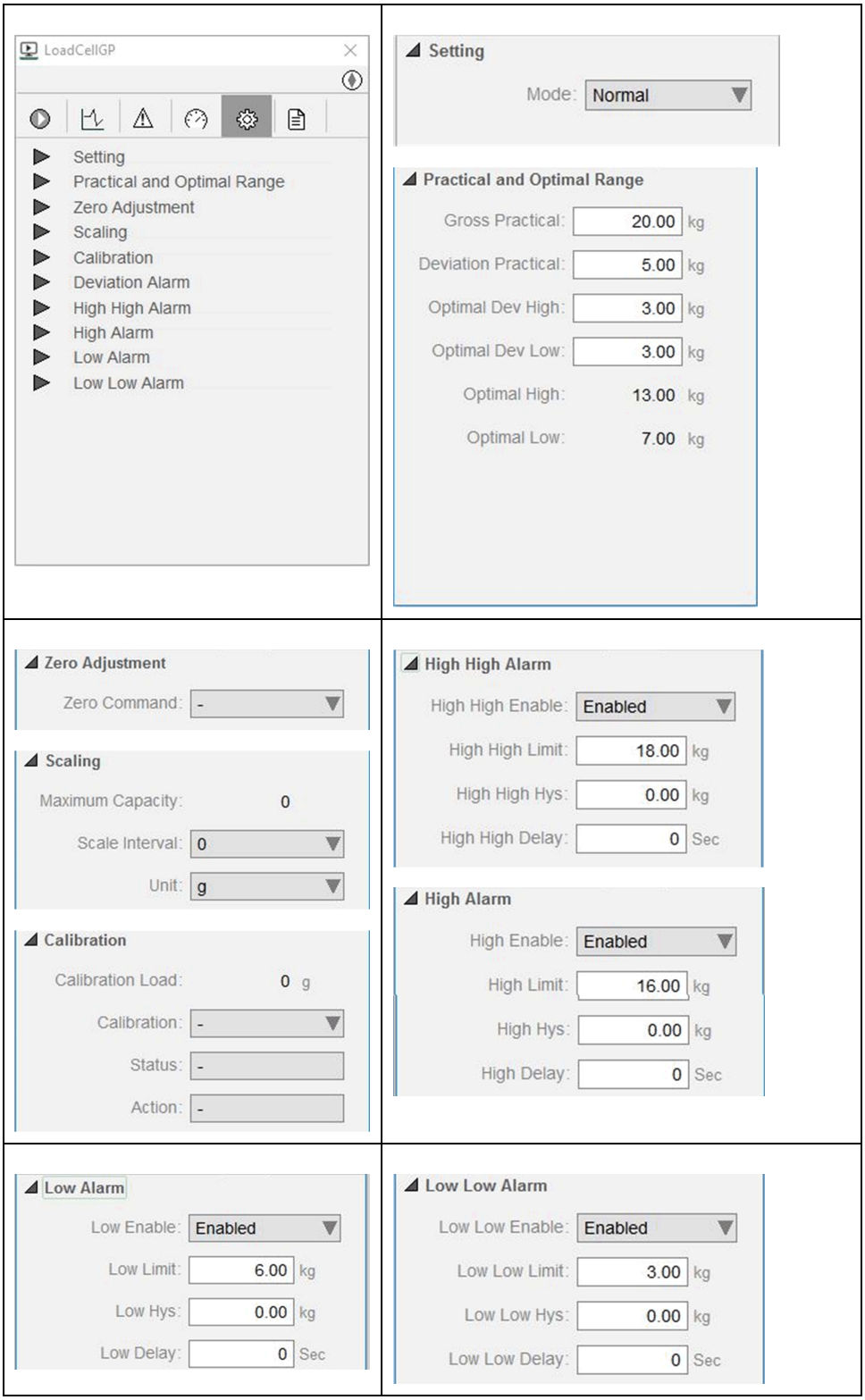
The table describes the control function of an operator tab:

Label	Item name	Description	Security parameter, page 577
Status	EquipStatus	Equipment status of loadcell. For more details, refer to EquipState calculated variable, page 587.	—
Current Owner	Owner	Current owner of loadcell, For more details, refer to operating owner, page 58.	—
Gross Weight	GrossPV	Gross measurement value in engineering unit.	—
Net Weight	NetPV	Net measurement value in engineering unit.	—
Net Setpoint	NetSP	Net setpoint in engineering unit.	Set Setpoint
Owner Command	OwnerSelect	Owner command of loadcell (Program or Operator).	Set Setpoint

Label	Item name	Description	Security parameter, page 577
Setpoint	Setpoint	Allows you to change the target setpoint, if owner = operator.	Set Setpoint
Tare	TarePV, TareCommand	Net measurement value in engineering unit and start tare and clear tare command.	Set Setpoint
—	Rearm	Allows you to reset manually after the detection of an inoperable device condition or not confirmed operation. **NOTE:** When the control module is reset, the current setpoint shown in operator tab is effective. Also, when you click Reset, a dialog box opens to confirm.	Set rearm

Engineering Tab

The table shows an engineering tab:



able describes the control function of engineering tab:

Label	Item name	Description	Security parameter, page 577
Settings			
Mode	Mode	Allows you to view and change the mode. For more details, refer to operating mode, page 59	Set parameters
Practical and optimal range			
Gross Practical	GrossPRRange	Practical range high gross view.	Set parameters
Deviation Practical	DeviationPR-Range	Deviation limit	

Label	Item name	Description	Security parameter, page 577
		High deviation range = GrossSP + DeviationPRRange Low deviation range = GrossSP - DeviationPRRange **NOTE:** This parameter is only applicable for deviation genie.	
Optimal Dev High	OptimalDev-HighSP	Optimal deviation high setpoint	
Optimal Dev Low	OptimalDev-LowSP	Optimal deviation low setpoint	
OR High	ORHigh	Optimal high limit	–
OR Low	ORLow	Optimal low limit	
Zero adjustment			
Zero Command	ZeroCommand	Zero adjustment command	Set parameters
Scaling			
Maximum capacity	MaxCapacity	Maximum capacity of calibrating module	Set parameters
Scale interval	ScaleInterval	Scale interval, Possible value are: 1, 2, 5, 10, 20, 50	
unit	CalibrationUnit	Engineering unit used for calibrating the module. Possible values are:g, kg, t, oz, lb, N	
Calibration			
Calibration load	CalibrationLoad	Calibration load for calibrate the module	Set parameters
Calibration	CalibrationOper-ation	Calibration command, possible options are: start, accept, cancel	
Status	StepCalibration	Current calibration status Number of the step calibration in progress. • 1 = Mismatch Param • 2 = Started • 3 = Zero Acquired. • 4 = Load Acquired • 5 = Done • 6 = Cancelled • 7 = Not Responding	–
Action	CalibrationAc-tion	Calibration action that needs to be performed	–
Step	ValidateStepCa-librationCom-mand	validate next step calibration command	Set parameters
Deviation Alarm			
High Deviation	HighDeviatio-nEnable	Enable high deviation	–
Low Deviation	LowDeviatio-nEnable	Enable low deviation	–
Deviation Hys	DeviationHyste-resis	Hysteresis value for deviation alarm.	Set parameters
Deviation SP	DeviationSet-point	Maximum deviation setpoint	Set setpoint

Label	Item name	Description	Security parameter, page 577
High Deviation SP	DeviationHigh-Setpoint	High deviation setpoint	–
Low Deviation SP	DeviationLowS-etpoint	Low deviation setpoint	–
Deviation Delay	DeviationDelay	Delay for enabling deviation	Set parameters
High Alarm			
High Enable	HighEnable	High alarm enable.	–
High Limit	HighLimit	High level setpoint.	Set setpoint
High Hys	HighLimitHyste-resis	Hysteresis high.	Set parameters
High Delay	HighLimitDelay	Delay high.	Set parameters
High High Alarm			
High High Enable	HighHighEnable	High-high alarm enable.	–
High High Limit	HighHighLimit	High-high level setpoint.	Set setpoint
High High Hys	HighHighLimi-tHysteresis	Hysteresis high-high.	Set parameters
High High Delay	HighHighLimit-Delay	Delay high-high.	Set parameters
Low Alarm			
Low Enable	LowEnable	Low alarm enable.	–
Low Limit	LowLimit	Low level setpoint.	Set setpoint
Low Hys	LowLimitHyste-resis	Hysteresis low.	Set parameters
Low Delay	LowLimitDelay	Delay low.	Set parameters
Low Low Alarm			
Low Low Enable	LowLowEnable	Low-low alarm enable.	–
Low Low Limit	LowLow Limit	Low-low level setpoint.	Set setpoint
Low Low Hys	LowLowLimi-tHysteresis	Hysteresis low-low.	Set parameters
Low Low Delay	LowLowLimitDe-lay	Delay low-low.	Set parameters

Measures Tab

The table lists the items of each groups of measures tab:

Group Order Number	**1**	**2**	**3**	**4**
Group name	**Measurement Status**	**Device Status**	**Response Status**	**Diagnostics**
Items	MeasureOK	DosingInProgress	Ready	WarningCode
	GrossPVError	CALIBDone	InProgress	FailCode0
	MeasureOOR	PhysicalCALIB	Completed	FailCode1
	AnalogOOR	TheoreticalCALIB	Error	FailCode2
	MeasureStable	WiringMode	–	–
	DecimalOOR	TerminateResistorSet	–	–
	MemoryFail	TheoreticalZeroDone	–	–
	PresetTareDone	TheoreticalCALIBDone	–	–

Group Order Number	1	2	3	4
Group name	**Measurement Status**	**Device Status**	**Response Status**	**Diagnostics**
	InputI0Enabled	ZeroAdustmentDone	–	–
	InputI1Enabled	SpanAdjustmentDone	–	–
	OutputS0Enabled	PhysicalCALIBDone	–	–
	OutputS1Enabled	ZeroAcquisitionDone	–	–
	OutputS2Enabled	Load1AcquisitionDone	–	–
	OutputS3Enabled	Load2AcquisitionDone	–	–
	TareDone	Load3AcquisitionDone	–	–
	ZeroTareDone	SaveDone	–	–

Items

Overview

This section describes the variables, equipstate and equipstatus variable, abnormal and action required conditions, alarm items, events, and trends item of `loadcell`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`UnknownState`	Unknown state	–	*Bit0*
`NotReady`	Not ready	–	*Bit1*
`DeviceFail`	Device is inoperable	–	*Bit3*
`Warning`	Alert state	–	*Bit3*
`CommunicationFail`	Communication interruption	–	*Bit4*
`RequireRearm`	Detected failure resetting required	Required/Not Required	*Bit5*
`ProgramSetpointMode`	Setpoint source for program owner	Remote/Sequence	*Bit6*
`Reseting`	Reset in progress	–	*Bit7*
`EnableDFB`	Enables the function	–	*Bit8*
`SignalFailure`	Signal detected failure	–	*Bit9*
`MeasurementStable`	Stability	–	*Bit10*
`CancelCalibration-Disable`	Disable the cancel calibration command	–	*Bit11*
`RearmEnable`	Enable the Reseting to reset the detected failure	–	*Bit12*
`ExtrenalCalibration`	Enable device calibration parameter values	–	*Bit13*

Item name	Description	Enumeration	Address
Bit items derived from ConfigurationWord.			
`Rearm`	Reset inoperable condition	Active/Inactive	*Bit0*
`OwnerSelect`	Owner of control module	Operator/Program	*Bit1*
`TareCommand`	Execution of tare command	–	*Bit2*
`CancelTareCommand`	Execution of cancel tare command	–	*Bit3*
`ZeroCommand`	Zero adjustment command	–	*Bit4*
`StartCalibration-Command`	Execution of physical calibration command	–	*Bit5*
`ValidateStepCali-brationCommand`	Validation step calibration	–	*Bit6*
`CancelCalibration-Command`	Execution of cancel current command	–	*Bit7*
`AcceptValues`	Accept calibrated new values	–	*Bit10*
`OutOfService`	Mode of operation	Out Of Service/ InService	*Bit15*
Bit items derived from ResponseStatus.			
`Ready`	Ready to accept a new command	–	*Bit0*
`InProgress`	Command execution in progress	–	*Bit1*
`Completed`	Command execution complete	–	*Bit2*
`Error`	Error during command execution	–	*Bit3*
Bit items derived from MeasurementStatus.			
`MeasureOK`	Measurement is OK	–	*Bit0*
`GrossPVError`	Gross weight is less than negative maximum capacity	–	*Bit1*
`MeasureOOR`	Gross weight is greater than maximum capacity	–	*Bit2*
`AnalogOOR`	Analog signal out of input range	–	*Bit3*
`MeasureStable`	Stability of measurement	–	*Bit4*
`DecimalOOR`	Measurement out of ¼ division flag	–	*Bit5*
`MemoryFail`	ROM Memory inoperable status	–	*Bit6*
`PresetTareDone`	Preset tare process flag	–	*Bit7*
`InputI0Enabled`	Logical Input I0 state flag	–	*Bit8*
`InputI1Enabled`	Logical Input I1 state flag	–	*Bit19*
`OutputS0Enabled`	Logical Output S0 (Dosing coarse feed) state flag	–	*Bit10*

Item name	Description	Enumeration	Address
OutputS1Enabled	Logical Output S1 (Dosing fine feed) state flag	–	*Bit11*
OutputS2Enabled	Logical Output S2 (Threshold) state flag	–	*Bit12*
OutputS3Enabled	Logical Output S3 (Threshold) state flag	–	*Bit13*
TareDone	Tare process flag	–	*Bit14*
ZeroTareDone	Zero/Tare process flag	–	*Bit15*
Bit items derived from DeviceStatus.			
DosingInProgress	Dosing cycle status	–	*Bit0*
CALIBDone	User calibration status	–	*Bit1*
PhysicalCALIB	Physical calibration is set	–	*Bit2*
TheoreticalCALIB	Theoretical calibration is set	–	*Bit3*
WiringMode	Load cell wiring mode: 0 = 4 wire, 1 = 6 wire	–	*Bit4*
TerminateResistor-Set	Termination resistor is set	–	*Bit5*
TheoreticalZeroDone	Theoretical zero done	–	*Bit6*
TheoreticalCALIB-Done	Theoretical calibration is done	–	*Bit7*
ZeroAdustmentDone	Zero adjustment done	–	*Bit8*
SpanAdjustmentDone	Span adjustment done	–	*Bit19*
PhysicalCALIBDone	Start physical calibration done	–	*Bit10*
ZeroAcquisitionDone	Zero acquisition done	–	*Bit11*
Load1Acquisition-Done	Load 1 acquisition done	–	*Bit12*
Load2Acquisition-Done	Load 2 acquisition done	–	*Bit13*
Load3Acquisition-Done	Load 3 acquisition done	–	*Bit14*
SaveDone	Save calibration done	–	*Bit15*
IO Items			
StatusWord	Status word	–	LoadCellGP_ST.STW
ConfigurationWord	Configuration word		LoadCellGP_ST.CFGW
Calculated Items			
Owner	Current owner of the loadcell. Refer to calculated variable, page 58.	–	GPL_Owner
Equipstatus	Current status of the loadcell. Refer to EquipStatus	–	GPL_LoadCellStatus

Item name	Description	Enumeration	Address
	calculated variable, page 587.		
Equipstate	Current state of the loadcell. Refer to EquipState calculated variable, page 587.	–	GPL_ LoadCellEquip-State
Abnormal	Shows abnormal condition of the loadcell. Refer to Abnormal Conditions, page 587.	–	GPL_ LoadCellAbnormal
Action required	Shows action required condition of the loadcell. Refer to Abnormal Conditions, page 587.	–	GPL_ ActionRequired
CalibrationOpera-tion	Caliberation command start, accept, and cancel.	–	GPL_ LoadCellCurrent-Calibration
PRHigh	When ViewSelection = 1 then (Tare PV+ NetSP + (2*Optimal Deviation High) and when ViewSelection = 0 (Gross Practical range).	–	GPL_ LoadCellPRHigh-Calc
PRLow	When ViewSelection = 1 (TarePV) and when ViewSelection = 0 (Zero)	–	GPL_ LoadCellPRLowCalc
PV	When ViewSelection = 1 then PV= NestPV and when ViewSelection = 0 then PV= GrossPV	–	GPL_ LoadCellPVValue
ORHigh	Optimal range high	–	GPL_ LoadCellORHigh-Calc
ORLow	Optimal range low	–	GPL_ LoadCellORLowCalc

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	Conditions	Enumeration
Passive with out of service	OOS	–
Passive	Not OOS x Ready	–
Active	Not OOS x Not Ready	–

EquipStatus Calculated Variable

The table describes the *EquipStatus* calculated variable:

Equipment Status	Conditions	Enumeration
Passive	`CommunicationFail`	Detected comm fail
Passive	Not `CommunicationFail` x `DeviceFail`	Detected device fail
Passive	Not `CommunicationFail` x Not `DeviceFail` x `OutOfService`	Out of service
Passive	Not `CommunicationFail` x Not `DeviceFail` x Not `OutOfService` x `Warning`	Detected warning
Passive	Not `CommunicationFail` x Not `DeviceFail` x Not `OutOfService` x Not `Warning` x `UnknownState`	Unknown state
Passive	Not `CommunicationFail` x Not `DeviceFail` x Not `OutOfService` x Not `Warning` x Not `UnknownState` x Not `Ready`	Inactive
Active	Not `CommunicationFail` x Not `DeviceFail` x Not `OutOfService` x Not `Warning` x Not `UnknownState` x `Ready`	Active

Abnormal Conditions

The table describes the abnormal conditions:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
`AbnormalOwner`	Abnormal owner status appears, if the current owner is not a normal owner which configured at the instance level.	*Bit0*
`AbnormalOutOfService`	Abnormal out of service status appears if the current operating mode is not normal mode.	*Bit1*
`AbnormalHighDeviatio-nEnable`	If low alarm enable is active.	*Bit4*
`AbnormalLowLowEnable`	If low-low alarm enable is active.	*Bit5*
`AbnormalLowEnable`	If low alarm enable is active.	*Bit6*
`AbnormalLowDeviatio-nEnable`	If low alarm enable is active.	*Bit7*
`AbnormalOperatorTab`	Abnormal status of operator tab. This condition appears if the Interlock bypass is enabled.	*Bit8*
`AbnormalHighHighEnable`	If high-high alarm enable is active.	*Bit9*
`AbnormalHighEnable`	If high alarm enable is active.	*Bit10*
`AbnormalEngineeringTab`	Abnormal status of engineering tab.	*Bit12*

Action Required Conditions

The table describes the Action required condition:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
`ActionOperatorTab`	Action required status of operator tab.	*Bit8*

Alarms

The table describes the advanced alarm items that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
ChannelFailure	Channel failure	GPL_BitValueCheck(StatusWord,0)
DeviceFailure	Device failure	GPL_BitValueCheck(StatusWord,3)
DeviceWarming	Device warning	GPL_BitValueCheck(StatusWord,4)
CommunicationFailure	Communication failure	GPL_BitValueCheck(StatusWord,5)

Alarms

Alternate Services:

- LAlarm, page 123
- DAlarm, page 128

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
StateAlarm	All changes in the loadcell status and output.
ModeAlarm	All changes in the mode and owner.
ActionAlarm	All actions by the operator.

Trends

The table describes the trend items that are used by Supervision components:

Item name	Expression	Description
GrossWeight	LoadCellGP_ST.GrossPV	Gross process value.
GrossSP	LoadCellGP_ST.SP	Gross Setpoint.
HighHighLimit	LALARMGP_CFG.SPHH	High-high level set Point.
HighLimit	LALARMGP_CFG.SPH	High level set Point.
LowLimit	LALARMGP_CFG.SPL	Low level set point.
LowLowLimit	LALARMGP_CFG.SPLL	Low-low level set point.

BMXERT1604

What's in This Chapter

Supervision.....516

Overview

This chapter describes the components that provide the functionality of template and Supervision functions of the BMXERT1604.

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the BMXERT1604.

Tags

Template

Refer to Special Cards chapter of ***EcoStruxure Process Expert Foundation Library Application Template User guide***.

Variable Tags

The following table describes the variable tags managed by the Supervision components:

Tag Name	Description
CM name_BMXERT_Var1	Channel value

Time Stamped Digital Alarm Tags

The following table describes the time stamped digital alarm tags managed by the Supervision components. It also indicates the corresponding variable tag for the digital alarms such that the alarm is activated based on the state of single or multiple digital variable tags:

Alarm Tag	Description	Variable Tag A
CM name_BMXERT_ TSDAlm	Time stamped alarm	*CM name_BMXERT_Var1*

NOTE: This time stamped digital alarm tag is created only when the channel is configured as a time stamping type channel.

Advanced Alarm Tags

The following table describes the advanced alarm tags managed by the Supervision components. It also indicates the expression that is configured in the

Supervision components to read or write the corresponding bit of the status or configuration word:

Alarm Tag	Description	Expression
CM name_BMXERT_AAAlm	Advanced alarm	*CM name_BMXERT_Var1*

NOTE: This advanced alarm tag is created only when the channel is configured as a discrete input type channel.

Trend Tags

No trend tags are managed by the Supervision components of BMXERT management.

Continuous Control

What's in This Part

PID 519
Ratio Controller Management 550
Ramp 564
Lead-Lag 576
Split-Range Controller 589
Step3 Controller 602
Internal Model Controller 617
Pulse Width Modulation Controller 638

Overview

This part provides the functionality of the templates, a detailed description of the control (functions,pins, pin layout, and variables of the function blocks), and the Supervision functions for the Continuous Control.

These components do not reflect any specific installation.

⚠ WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate them.
- Review the implications of communication link interruptions and take actions to mitigate them.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, and applicable codes and regulations.
- Apply local accident prevention and safety regulations and guidelines.[1]
- Test each implementation of a system for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

PID

What's in This Chapter

Template 519
Control - PIDCTLGP 524
Control - PIDMUXGP 529
Control - PIDTUNEGP 532
Supervision 536

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the PID with monitoring interface.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$PIDGP` template.

This process template helps monitor and control the controller from the Supervision system and provides the operating modes used in the remaining EcoStruxure Process Expert process templates

Description

General Description

The `$PIDGP` process control module template is used to condition the signals associated with the control of a PID Feed Forward (PIDFF)-type controller.

Function Description

The main functions of the PID template are described in the following table:

Function	Description
PIDFF	The template includes and incorporates functions provided by the PIDFF controller from the standard `CONTR_CTL` library. You can directly configure the PIDFF controller functions that have not been shown through the template interface by accessing its configuration parameters (for example, FeedForward, variable gain, and so on).
Owner	The object manages control system level, which is the owner (Operator or Program). As a result, it is responsible for setting the set-point to the desired position (in the Auto mode of the PID controller) or the manual output (in the Manual mode of the PID controller).
Interlocking	The template enables to move to the defined position if an active interlock that requires this move is detected. An interlock bypass function is available.
Local and remote set-point	Enables to control the template with a set-point that is determined through a set-point selector (local or remote) in program owner. The local set-point is assigned from the sequential control while the remote set-point is assigned to the template from the logic implemented in the continuous control.
Override	You can combine several `PIDCTLGP` or other templates to control one single control module. This way, the template that is not controlling this template at a certain point in time works with an external output, that is, calculating the

Function	Description
	new output based on the position of control module instead of the last position calculated by the template itself. These functionalities are internally provided by the standard PIDFF controller.
Auto tune	It provides an option to tune PID gains based on the estimated response.
Multiplexer	The PIDMUXGP template is used to multiplex 2 different parameter configurations that affect a single PID controller. This way, you can use one single PID controller with different configurations (for example, split-range PID for cooling or heating).
Alarms	It provides an option to detect timed alarms associated with an analog signal. The template provides detected alarm functions by level (very high, high, low and/or very low) and by deviation in relation to a set-point value.

Optional Services

Available optional service:

- Interlock

Parameters

Parameters

The `$PIDGP` template defines operation of a typical PID. As a feedback controller, it delivers the control output at desired levels.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default value	Additional remarks
Range	**Output high range (EU)**	`Float`	*100.0*	High range of output
	Output low range (EU)		*0.0*	low range of output
	Output engineering units	`String`	*%*	Engineering units of output
	Output format		*####.#*	Format type of output
	Output high limit range	`Float`	*100.0*	–
	Output low limit range		*0.0*	–
Setpoint	**Force remote setpoint**	`Boolean`	*FALSE*	Enables the remote control operation.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

AAlarm

For parameters of LimitAlarm, page 118 and DeviationAlarm, page 124.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default value	Additional remarks
Operation	**Normal Owner**	Enum	*Program and Cascade (5)*	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	**Set Owner**	Enum	*Operator (1)*	Privilege to change owner. This parameter is used in Operator Tab, page 540.
	Set Setpoint		*Operator (1)*	Privilege to change setpoint. This parameter is used in Operator Tab, page 540.
	Set Reset		*Operator (Confirmed) (10)*	Privilege to reset. This parameter is used in Operator Tab, page 540.
	Set Parameters		*Engineer (5)*	Privilege to set parameters. This parameter is used in Engineering Tab, page 541.
	Set Alarm Limits		*Operator (1)*	Privilege to set alarm limits. This parameter is used in Operator Tab, page 540.
	Alarm Acknowledge		*Operator (1)*	Privilege to acknowledge alarm. This parameter is used in Alarm Tab, page 67.
	Set Bypass Conditions		*Supervisor (2)*	Privilege to set bypass conditions mode. This parameter is used in Engineering Tab, page 541.
	Set Bypass All Interlocks		*Supervisor (2)*	Privilege to bypass all interlocks. This parameter is used in Engineering Tab, page 541.
Trend	For more details, refer to the topic describing the parameters of trend, page 47.			
Data				
Configuration	**Normal mode**	Enum	*Auto (0)*	–
Range	**Optimal range**	Enum	*Disable (0)*	It disables the optimal range indication.
	Practical range high/low disable	Boolean	*TRUE*	It disables practical range high and practical low value.
	Practical range disable		*TRUE*	It disables the practical range indication.
Deaband	**Present value/ setpoint**	Integer	*0*	Deadband value for present value.
	Output			Deadband value for output.
Trend	**Disable present value**	Boolean	*FALSE*	–
	Disable setpoint			–
	Disable output			–
Events	**Enable log**	Boolean	*0*	Enable event logging for asset status.
Historize	**Gain/KD/TI/TD**	Boolean	*FALSE*	–
	Present value			It allows to historize the change present value.
	Setpoint			It allows to historize the change local setpoint.
	Output			–
AutoTune				

Section	Parameter description	Type	Default value	Additional remarks
Events	**Enable log**	`Boolean`	*0*	Enable event logging for asset status.
Historize	**Autotune time/ step**	`Boolean`	*FALSE*	–
Multiplexer				
Configura-tion	**Tuning parameter 1**	`String`	*PID 1*	–
	Tuning parameter 2	`String`	*PID 2*	–
Historize	**Output**	`Boolean`	*FALSE*	–
	Gain	`Boolean`	*FALSE*	–
	Integral time	`Boolean`	*FALSE*	–
	Derivative time	`Boolean`	*FALSE*	–
	Derivative gain	`Boolean`	*FALSE*	–

Asset and Display Parameters

For more details, refer to the topic describing the parameters of asset and display, page 43.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

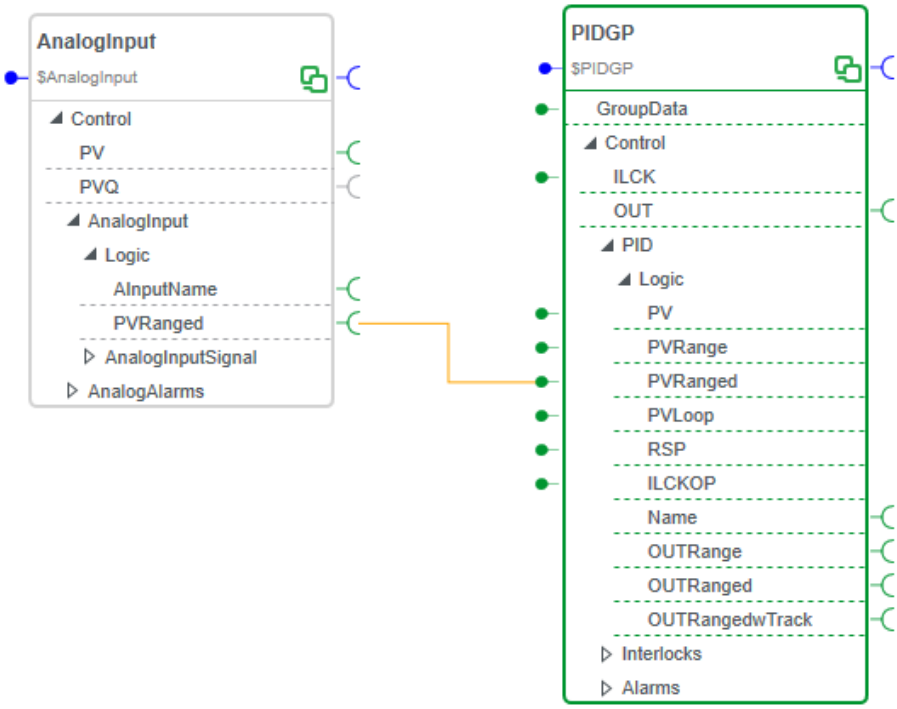
Refer for the parameters of LimitAlarm, page 118 and DeviationAlarm, page 124.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

The following figure depicts the `$PIDGP` (also applicable for `$PIDGP`) template as it appears in `Links Editor` and an example of other templates connected to it:



NOTE: The PVLoop interface has to be linked to the PID if the user wants to navigate from the PID faceplate to the analog input faceplate in supervision.

NOTE: For analog input and PID faceplate navigation, define the parameter `Process Page Name` where these instances will be located in graphic page else during navigation the process display area will not show the content. For parameter description, refer to, page 43.

The following table lists the link interface that creates link symbol on faceplate:

Interface name	Runtime parameter
`PV`	*PresentValueNavigate*
`PVRange`	
`PVRanged`	
`RSP`	*SetpointNavigate*
`OUTRange`	*OutputNavigate*
`OUTRanged`	
`OUTRangedwTrack`	

For more details, refer to Supervision chapter > Faceplate > Navigation Link of General Purpose Library User Guide.

Refer to common control module template interface, page 47 for more details on common interfaces.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control - PIDCTLGP

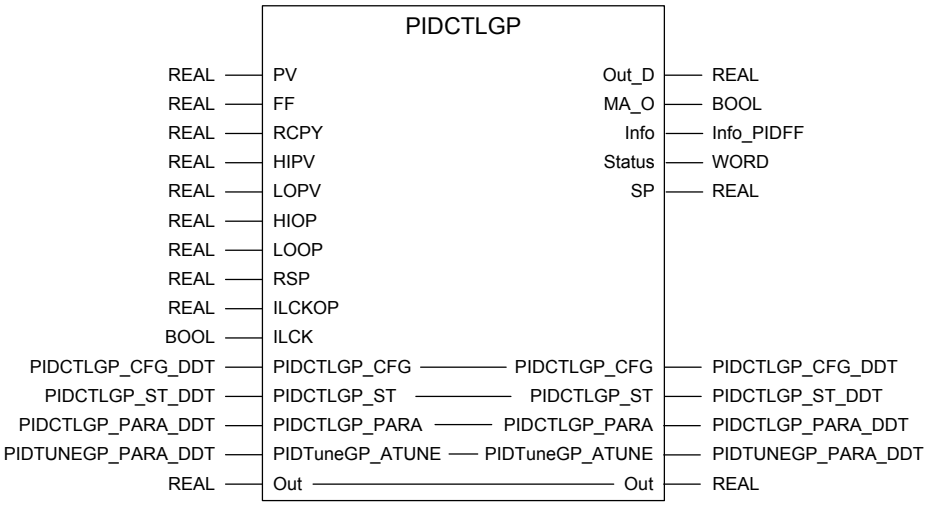
Overview

This section describes the `PIDCTLGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
`PV`	`REAL`	Current value of the measurement in engineering units.
`FF`	`REAL`	Value of the variable that disrupts the behavior of the measurement. This input is internally connected to the FF input of the PIDFF controller. By default, a disruption value does not have any repercussion on the output. Refer to the `PARA.ff_inf`, `PARA.ff_sup`, `PARA.otff_inf`, and `PARA.otff_sup` parameters of a PIDFF controller. These values are initialized to zero by default.
`RCPY`	`REAL`	Value of the output current (actual) status. You can use this input to feed back the output in case of multiple controllers working against the same output (Override). This input is internally connected to the `RCPY` input of the PIDFF controller. Refer to the `PARA.EN_RCPY` input.
`HIPV`	`REAL`	Upper value of the measurement range. Used for limiting the maximum set-point admitted by the controller. This input is internally connected to the `PARA.PV_SUP` parameter of the PIDFF controller (therefore, you cannot set this parameter from outside the `PIDCTLGP` DFB because it would be overwritten).
`LOPV`	`REAL`	Lower value of the measurement range. Used for limiting the minimum set-point admitted by the controller. This input is internally connected to the `PARA.PV_INF` parameter of the PIDFF controller (therefore, you cannot set this parameter from outside the `PIDCTLGP` DFB because it would be overwritten).
`HIOP`	`REAL`	Upper value of the output range. Used for limiting the maximum output generated by the controller. This input is internally connected to the `PARA.OUT_SUP` parameter of the PIDFF controller (therefore, you cannot set this

Parameter	Type	Description
		parameter from outside the PIDCTLGP DFB because it would be overwritten).
LOOP	REAL	Lower value of the output range. Used for limiting the minimum output generated by the controller. This input is internally connected to the PARA.OUT_INF parameter of the PIDFF controller (therefore, you cannot set this parameter from outside the PIDCTLGP DFB because it would be overwritten).
RSP	REAL	Remote set-point. Usually set by the continuous control; for example, by a PID controller output. Operates in Program/Casc mode (Automatic + Remote). Refer to the Rem, page 528.
ILCKOP	REAL	Use this output when the DFB is interlocked. Refer to the ILCK input.
ILCK	BOOL	1 = Interlocks the controller at a defined position with the ILCKOP input. Internally, the PIDFF controller is set to Tracking mode.

Outputs

Output Parameter Description

Parameter		Type	Description
OUTD		REAL	Differential output. Is generated based on the OUTD output of the PIDFF controller.
MA_O		BOOL	1 = Current operating mode. Is generated based on the MA_O output of the PIDFF controller. Refer to PIDFF controller for more details.
INFO	–	Info_PIDFF	Variables output by the PIDFF controller on the INFO output. Refer to the PIDFF controller for more details.
	dev	REAL	Deviation value (PV - SP).
	out_ff	REAL	Value of the feed forward contribution.
Status		WORD	Status output by the PIDFF controller on the Status output. Refer to the PIDFF controller for more details.
SP		REAL	Current set-point that is being considered by the PID algorithm.

The set-point calculation that the DFB performs based on the signal status is detailed in the following table:

Owner (OFF: Program; ON: Operator	Rem	SP is calculated as:
OFF	OFF	SC.LSP
OFF	ON	RSP
ON	-	PIDCTLGP_ST.SP You can modify the set-point directly from the monitoring subsystem while this mode is active.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
PIDCTLGP_CFG	PIDCTLGP_CFG_DDT	Provides the data necessary to configure the DFB from the monitoring subsystem or the sequential control.
PIDCTLGP_ST	PIDCTLGP_ST_DDT	Provides the data necessary to monitor and/or control the DFB status.
PIDCTLGP_PARA	PIDCTLGP_PARA_DDT	Data structure of the PIDFF controller configuration within the PIDCTLGP DFB. Refer standard PIDFF DFB in the Control documentation for more details.
PIDTuneGP_ATUNE	PIDTUNEGP_PARA_DDT, page 533	Data structure of the AUTOTUNEGP DFB configuration within the PIDCTLGP DFB.
Out	REAL	Output signal of the controller. Refer standard PIDFF DFB in the Control documentation for more details.

PIDCTLGP_CFG_DDT Type

Name	Type	Description
OPHILimit	REAL	Enables to define the maximum value of the output for the Automatic operation of the controller. This input is internally connected to the PARA.OUT_MAX parameter of the PIDFF controller (therefore, you cannot set the parameter from outside the PIDCTLGP DFB because it would be overwritten).
OPLOLimit	REAL	Enables to define the minimum value of the output for the Automatic operation of the controller. This input is internally connected to the PARA.OUT_MIN parameter of the PIDFF controller (therefore, you cannot set the parameter from outside the PIDCTLGP DFB because it would be overwritten).
SPHILimit	REAL	Enables to define the maximum value of the setpoint for the Automatic operation of the controller. This input is internally connected to the PARA.SP_MAX parameter of the PIDFF controller (therefore, you cannot set the parameter from outside the PIDCTLGP DFB because it would be overwritten).
SPLOLimit	REAL	Enables to define the minimum value of the setpoint for the Automatic operation of the controller. This input is internally connected to the PARA.SP_MIN parameter of the PIDFF controller (therefore, you cannot set the parameter from outside the PIDCTLGP DFB because it would be overwritten).
KD	REAL	Enables to set the derivative gain parameter of the controller. This input is internally connected to the PARA.KD parameter of the PIDFF controller (therefore, you cannot set the parameter from outside the PIDCTLGP DFB because it would be overwritten).
Gain	REAL	Enables to set the proportional gain parameter of the controller. This input is internally connected to the PARA.KP parameter of the PIDFF controller (therefore, you cannot the parameter from outside the PIDCTLGP DFB because it would be overwritten).
TI	TIME	Enables to set the integral time parameter of the controller. This input is internally connected to the PARA.TI parameter of the PIDFF controller (therefore, you cannot set the parameter from outside the PIDCTLGP DFB because it would be overwritten).
TD	TIME	Enables to set the derivative time parameter of the controller. This input is internally connected to the PARA.TD parameter of the PIDFF controller (therefore, you cannot set the parameter from outside the PIDCTLGP DFB because it would be overwritten).

PIDCTLGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
PV	–	–	REAL	Read-only access This variable indicates what the current measurement is.
SP	–	–	REAL	Read/write access Setpoint This value indicates the setpoint actually considered as Controller's setpoint.
OP	–	–	REAL	Read/write access Output
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	Rem	*Bit0*	BOOL	Refer to the Rem, page 528.
	ILCK	*Bit2*	BOOL	Refer to the ILCK input pin, page 524.
	OVRide	*Bit3*	BOOL	Determines whether the DFB is operating with external output (1 – Override) or not (0 – Normal). Refer to the PARA.EN_RCPY input.
	AutoIn-valid	*Bit4*	BOOL	1 = Indicates that the configuration is invalid.
	Warning	*Bit5*	BOOL	1 = Indicates an alert
	LowThr	*Bit6*	BOOL	1 = The output OUT has reached the lower threshold out_min.
	UpperThr	*Bit7*	BOOL	1 = The output OUT has reached the upper threshold out_max.
	ThrIND	*Bit8*	BOOL	1 = The thresholds pv_inf and pv_sup are identical.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
	Owner	*Bit0*	BOOL	Read/write access Enables to configure whether the set-point is set by the Program (0) or the Operator (1).
	ILCKBP	*Bit1*	BOOL	Read/write access Enables the interlocking to be bypassed (1).
	Mode	*Bit2*	BOOL	Read/write access Reports the operating mode of the PIDFF controller if the owner is the Program, and enables the operating mode to be configured if the owner is the Operator. 1 corresponds to Auto and 0 to Manual.
	ACT	*Bit3*	BOOL	Read/write access Enables the PIFF controller action to be configured. 0 corresponds to reverse and 1 to forward. The DFB automatically assigns the value of this signal to the PARA.rev_dir signal of the IMC controller .
	PVDEV	*Bit4*	BOOL	Read/write access Enables to configure whether the derivative action applies only to the development of the measurement (PV) or the detected error (PV-SP or SP-PV depending on the configured action). 0 corresponds to considering the measurement only, and 1 to the detected error. The DFB automatically assigns the value of this signal to the PARA.pv_dev signal of the PIDFF controller.
	MixPar	*Bit5*	BOOL	Read/write access Enables to configure whether the PIDFF controller calculation formula is mixed (proportional gain affects integral and derivative terms) or parallel (the action only affects the proportional term). 0 corresponds to mixed and 1 to parallel. The DFB automatically assigns the value of this signal to the PARA.MIX_PAR signal of the PIDFF controller.

PIDCTLGP_PARA_DDT Type

Name	Bit Name	Type	Description
PARA_PIDFF	–	Para_PIDFF	PIDFF parameters allows to populate additional data for PIDFF.
	id	UINT	Reserved for autotuning.
	pv_inf	REAL	Lower limit of the process value range.
	pv_sup	REAL	Upper limit of the process value range.
	out_inf	REAL	Lower limit of the output value range.
	out_sup	REAL	Upper limit of the output value range.
	rev_dir	BOOL	Parameter block for PIDFF: • 0 = opposite action of the PID controller. • 1 = direct action of the PID controller.
	mix_par	BOOL	• 1 = PID controller with parallel structure. • 1 = PID controller with mixed structure.
	aw_type	BOOL	1 =Anti windup halt is filtered.
	en_rcpy	BOOL	1= The RCPY input is used.
	kp	REAL	Proportional action coefficient (gain).
	ti	TIME	Integral time
	td	TIME	Derivative time
	kd	REAL	Differential gain
	pv_dev	BOOL	Type of differential contribution: • 1 = Differential contribution in relation to system deviation. • 0 = Differential contribution in relation to regulating variable (process value).
	bump	BOOL	• 1 = Transition to automatic mode with bump. • 0 = Bumpless transition to automatic mode.
	dband	REAL	Dead zone on deviation.
	gain_kp	REAL	Reducing the proportional contribution within the dead zone dband.
	ovs_att	REAL	Reducing the overrun.
	outbias	REAL	Manual compensation for the static deviation.
	out_min	REAL	Lower limit of the output.
	out_max	REAL	Upper limit of the output.
	outrate	REAL	Limit for output modification in units per second (= 0).
	ff_inf	REAL	Lower limit of the FF range.
	ff_sup	REAL	Upper limit of the FF range.
	otff_inf	REAL	Lower limit of the out_FF range.
	otff_sup	REAL	Upper limit of the out_FF range.
SAMPLETM	–	TIME	Sample time for connected control SAMPLETM controller configuration within the PIDCTL DFB. Refer to the standard SAMPLETM DFB in the *Control* documentation for more details.

Public Variables

Public Variable Description

Variable	Type	Description
SC	PIDCTLGP_SC_DDT	Provides the frequently needed data to monitor and control the regulator status from the sequential control.

PIDCTLGP_SC_DDT Type

Name	Type	Description
LSP	REAL	Read/write access Enables the assignment of the local set-point for the sequential control if the owner is the Program (Owner is 0), the selected mode is AUTO, and the selected set-point is Local (SC.Rem is 0). Otherwise, the current set-point (SP output) is continuously copied to this variable.
LOP	REAL	Read/write access Enables the assignment of the local output for the sequential control if the owner is the Program (Owner is 0) and the mode is MANUAL. Otherwise, the current output (OP output) is continuously copied to this variable.
PV	REAL	Read-only access Refer to the PV input pin, page 524.
Owner	BOOL	Read-only access Refer to the Owner input/output pin, page 526.
Man	BOOL	Write-only access 1 = Allows the controller to be changed to Manual mode if the owner is the Program. The DFB automatically sets the signal to 0 after processing the signal.
Auto	BOOL	Write-only access 1 = Allows the controller to be switched to Auto mode (if the owner is the Program. The DFB automatically sets the signal to 0 after processing the signal.
Mode	BOOL	Read-only access 1 = Auto 0 = Manual Refer to the PIDCTLGP_ST.CFGW. Mode input/output pin, page 526.
Rem	BOOL	Read/write access 1 = Allows the DFB to be configured for a remote set-point -RSP, if the owner is the Program and the mode is Auto. 0 = Allows the DFB to be configured for a local set-point -LSP, if the owner is the Program and the mode is Auto.
ILCKD	BOOL	Read-only access The signal evaluation depending on the ILCK input and the PIDCTLGP_ST.STW.ILCKBP input/output is shown in the following table.

ILCKD	ILCKBP	SC.ILCKD
OFF	-	OFF
ON	OFF	ON
ON	ON	OFF

Control - PIDMUXGP

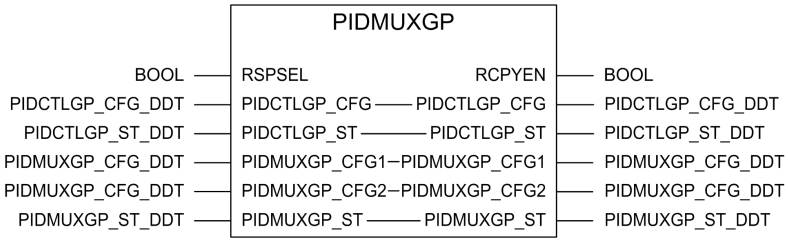
Overview

This section describes the PIDMUXGP DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
RSPSEL	BOOL	Remote set point for selecting the parameter set normally set by the continuous control. Operates in Program /Casc mode (Automatic + Remote). Refer to the Rem, page 532. 0 = Corresponds to select parameter configuration 1 (PIDMUXGP_CFG1) 1 = Corresponds to select parameter configuration 2 (PIDMUXGP_CFG2).

Outputs

Output Parameter Description

Parameter	Type	Description
RCPYEN	BOOL	1 = Activates output during a scan cycle when the parameter configuration change is made. This output can be used to override the PIDCTLGP controller (PIDCTLGP.RCPY input) for which the parameter configuration change is being made. The change takes place without any jumps occurring on the output.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
PIDCTLGP_CFG	PIDCTLGP_CFG_DDT, page 526	The configuration data structure belonging to the PIDCTLGP module for which the parameters need to be configured (PIDCTLGP.PIDCTLGP_ST input/output) needs to be connected to this input. Refer to the PIDCTLGP module in the PIDCTLGP module user manual for more details.
PIDCTLGP_ST	PIDCTLGP_ST_DDT, page 527	The status data structure belonging to the PIDCTLGP module for which the parameters need to be configured (PIDCTLGP.PIDCTLGP_ST input/output) needs to be connected to this input. Refer to the PIDCTLGP module in the PIDCTLGP module user manual for more details.

Parameter	Type	Description
PIDMUXGP_CFGx	PIDMUXGP_CFG_DDT	Provides the data needed to configure the configuration parameters for the associated PIDCTLGP module. This data can be accessed from the monitoring subsystem.
PIDMUXGP_ST	PIDMUXGP_ST_DDT	Provides the data needed to monitor and/or control the DFB status.

PIDMUXGP_CFGx_DDT Type

Name	Bit Name	Bit	Type	Description
CFGW	–	–	WORD	Read/write access Configuration word belonging to the associated PIDCTLGP controller.
	ACT	Bit3	BOOL	Enables the PIFF controller action to be configured: • 0 = Reverse • 1 = Direct the DFB automatically assigns the value of this signal to the PARA.rev_dir signal of the IMC controller
	PVDEV	Bit4	BOOL	Enables to configure whether the derivative action applies only to the development of the measurement (PV) or the detected error (PV-SP or SP-PV depending on the configured action). 0 corresponds to considering the measurement only, and 1 to the detected error. The DFB automatically assigns the value of this signal to the PARA.pv_dev signal of the PIDFF controller.
	MixPar	Bit5	BOOL	Enables to configure whether the PIDFF controller calculation formula is mixed (proportional gain affects integral and derivative terms) or parallel (the action only affects the proportional term). 0 corresponds to mixed and 1 to parallel. The DFB automatically assigns the value of this signal to the PARA.MIX_PAR signal of the PIDFF controller.
OPHILimit	–	–	REAL	Enables to define the maximum value of the output for the Automatic operating mode of the controller.
OPLOLimit	–	–	REAL	Enables to define the minimum value of the output for the Automatic operating mode of the controller.
Gain	–	–	REAL	Enables to set the proportional gain parameter of the controller.
TI	–	–	TIME	Enables to set the integral time parameter of the controller.
TD	–	–	TIME	Enables to set the derivative time parameter of the controller.
KD	–	–	REAL	Enables to set the derivative gain parameter of the controller.

PIDMUXGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	Rem	Bit0	BOOL	Refer to the Rem, page 532.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
	Owner	Bit0	BOOL	Read/write access Enables to configure whether the set-point is set by the Program (0) or by the Operator (1).

Name	Bit Name	Bit	Type	Description
	LSPSEL	Bit1	BOOL	Read/write access This variable indicates which parameter configuration is selected (0: parameter configuration 1; 1: parameter configuration 2) and is calculated internally in the DFB as long as the Operator is not the owner. In this case, it can be modified from the monitoring subsystem.

Public Variables

Public Variable Description

Variable	Type	Description
SC	PIDMUXGP_SC_DDT	Provides the frequently needed data to monitor and control the module status from the sequential control.

PIDMUXGP_SC_DDT Type

Name	Type	Description
LSPSEL	BOOL	Read/write access 1 = Enables the sequential control to select the parameter configuration if the owner is the Program/Local (Owner and SC.Rem set to 0). 0 = The current selection set-point (SP output) is continuously copied to this variable.
Owner	BOOL	Read-only access 1 = Refer to the PIDMUXGP_ST.CFGW.Owner input/output pin, page 530.
Rem	BOOL	Read/write access 1 = Enables to configure the DFB for remote parameter configuration selection -RSP while the owner is the Program. 0 = Enables to configure the DFB for local parameter configuration selection -LSP while the owner is the Program.

Control - PIDTUNEGP

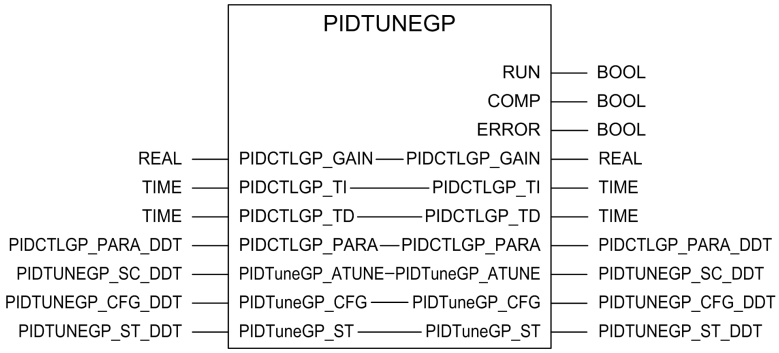
Overview

This section describes the PIDTUNEGP DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Outputs

Output Parameter Description

Parameter	Type	Description
RUN	BOOL	1 = Indicates the Running status of Autotune.
COMP	BOOL	1 = Indicates the Completion/Success of tunning.
ERROR	BOOL	1 = Indicates the ERROR status extracted from the AUTOTUNE DFB Status output pin For more details, refer to the AUTOTUNE *Control* DFB chapter from the *Control Expert* document.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
PIDCTLGP_GAIN	REAL	Refer to Gain, page 526.
PIDCTLGP_TI	TIME	Refer to TI, page 526.
PIDCTLGP_TD	TIME	Refer to TD, page 526.
PIDCTLGP_PARA	PIDCTLGP_PARA_DDT, page 528	Data structure of the PIDFF controller configuration within the PIDCTLGP DFB. Refer standard PIDFF DFB in the Control documentation for more details.
PIDTuneGP_ATUNE	PIDTUNEGP_PARA_DDT	Provides the means to control the device from the PID Public Autotune control.
PIDTuneGP_CFG	PIDTUNEGP_CFG_DDT	Provides the means to control the device from the monitoring subsystem.
PIDTuneGP_ST	PIDTUNEGP_ST_DDT	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory.

PIDTUNEGP_PARA_DDT Type

Name	Type	Description
aEN	BOOL	1 = Enable of the Autotune functionality. This input is internally connected to the EN input of the AUTOTUNE controller.
aPV	REAL	Process value of the measurement in engineering units. This input is internally connected to the to the PV input of the AUTOTUNE controller. For more details, refer to the AUTOTUNE.PV input pin description from the *control expert* document.

Name	Type	Description
aSP	REAL	Setpoint. This input is internally connected to the to the SP input of the AUTOTUNE controller. For more details, refer to the AUTOTUNE.SP input pin description from the *control expert* document.
aRcpy	REAL	Value of the output current (actual) status. You can use this input to provide the output feedback in case of multiple controllers are working against the same output (Override). This input is internally connected to the to the RCPY input of the AUTOTUNE controller. For more details, refer to the AUTOTUNE.RCPY input pin description from the *control expert* document.
aTRI	REAL	Start Input This input is internally connected to the to the TR_I input of the AUTOTUNE controller. For more details, refer to the AUTOTUNE.TR_I input pin description from the *control expert* document.
aTRS	BOOL	Start command This input is internally connected to the to the TR_S input of the AUTOTUNE controller. For more details, refer to the AUTOTUNE.TR_S input pin description from the *control expert* document.
pPV	REAL	PV value of PID DFB. This input is internally connected to the to the PV output of the AUTOTUNE controller. For more details, refer to the AUTOTUNE.PV output pin description from the *control expert* document.
pSP	REAL	SP value of PID DFB. This input is internally connected to the to the SP output of the AUTOTUNE controller. For more details, refer to the AUTOTUNE.SP output pin description from the *control expert* document.
pTRI	REAL	TRI value of PID DFB. This input is internally connected to the to the TRI output of the AUTOTUNE controller. For more details, refer to the AUTOTUNE.TRI output pin description from the *control expert* document.
pTRS	BOOL	TRS value of PID DFB. This input is internally connected to the to the TRS output of the AUTOTUNE controller. For more details, refer to the AUTOTUNE.TRS output pin description from the *control expert* document.
RUN	BOOL	Refer to the RUN output pin, page 533.
COMP	BOOL	Refer to the COMP output pin, page 533.

PIDTUNEGP_CFG_DDT Type

Name	Type	Description
GAIN	REAL	Enables to set the proportional gain parameter of the controller.
TI	TIME	Enables to set the integral time parameter of the controller.
TD	TIME	Enables to set the derivative time parameter of the controller.
PKP	REAL	Autotune previous proportional constant.

Name	Type	Description
PTI	TIME	Autotune previous integration time.
PTD	TIME	Autotune previous derivative time.
TUN-TIME	TIME	Autotune time.
TUN-SAMP	REAL	Autotune step amplitude.

PIDTUNEGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
CFGW	–	–	WORD	Provides the means to control the device from the monitoring (HMI) system. Read-only access to the data contained in this word.
	START	*Bit0*	BOOL	1 = Start autotune command.
	STOP	*Bit1*	BOOL	1 = Stop autotune command.
	TRANS	*Bit2*	BOOL	1 = Transfer parameters.
	RETURNP	*Bit3*	BOOL	1 = Return previous parameters.
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	ERRORF	*Bit0*	BOOL	1 = An error is detected in a floating point value calculation.
	INVALIDV	*Bit1*	BOOL	1 = Invalid value recorded at one of the floating point inputs.
	DIVISION0	*Bit2*	BOOL	1 = Division by zero with calculation in floating point values.
	CAPOVER	*Bit3*	BOOL	1 = Capacity overflow during floating point value calculation.
	PAROUT	*Bit4*	BOOL	1 = The parameter perf is outside the [0.1] range: the function block uses the value 0 or 1 for calculations.
	THRESHOLD	*Bit7*	BOOL	1 = The thresholds pv_inf and pv_sup of the controller to be set.
	PARACON	*Bit8*	BOOL	1 = The PARA_C output is not connected to the parameters of an autotunable controller.
	FAILED	*Bit9*	BOOL	1 = Autotuning is unsuccessful.
	SUCCES	*Bit10*	BOOL	1 = The last autotune was successful.
DIAGW	–	–	WORD	Provides the device diagnosis status.
	RUNNING	*Bit0*	BOOL	1 = Autotuning is running.
	ABORTED	*Bit1*	BOOL	1 = Autotuning aborted.
	PARERROR	*Bit2*	BOOL	1 = Parameter error is detected.
	ALTPAR	*Bit3*	BOOL	1 = Alteration of parameters, which have just been set automatically.
	STOPSERR	*Bit4*	BOOL	1 = Stop as a consequence of detected system error.
	SATURATED	*Bit5*	BOOL	1 = Process value saturated.
	ALTSMALL	*Bit6*	BOOL	1 = Alteration too small.
	SAMPINTER	*Bit7*	BOOL	1 = Sampling interval invalid.
	INCOMPRE	*Bit8*	BOOL	1 = Incomprehensible reaction.
	NONSTAB	*Bit9*	BOOL	1 = Non-stabilized measuring at the start.
	TMAXSHORT	*Bit10*	BOOL	1 = Length of actuating pulse Stellimpulses (tmax) too short.
	NOISE	*Bit11*	BOOL	1 = Too much noise/interference.
	TMAXLONG	*Bit12*	BOOL	1 = Length of actuating pulse (tmax) too long.
	EXTHRESH	*Bit13*	BOOL	1 = Process with significant exceeding of the thresholds.
	MINPHASE	*Bit14*	BOOL	1 = Process without minimum phase.

Name	Bit Name	Bit	Type	Description
	ASYPROCESS	*Bit15*	BOOL	1 = Asymmetric process.
NOTE: For more detials on diagnostic Information, refer to *control expert* documentation of AUTOTUNE DFB.				

Supervision

Overview

This section describes the Supervision functions and runtime services that are available for the PID.

Genies

Genie Properties

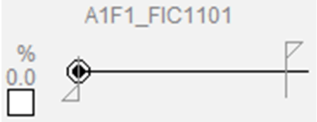
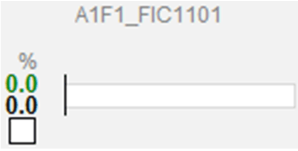
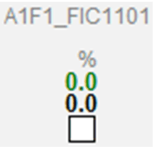
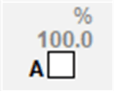
Refer to Using Genies, page 62.

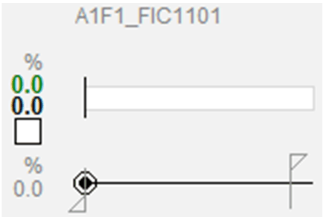
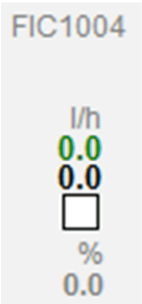
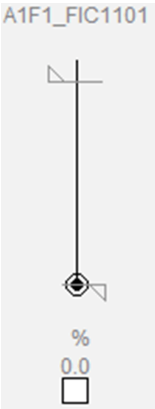
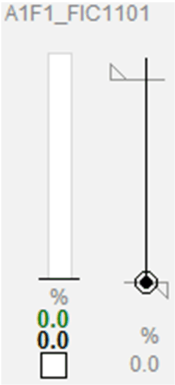
Genie Icons

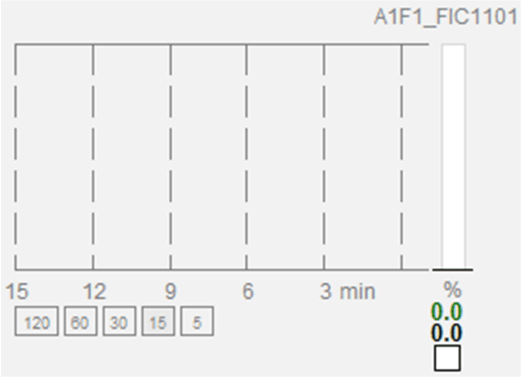
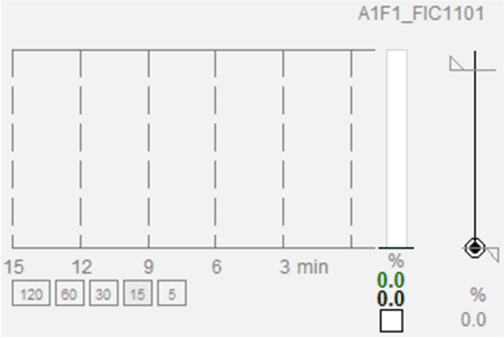
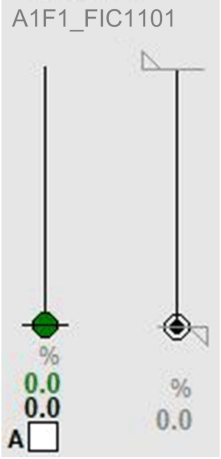
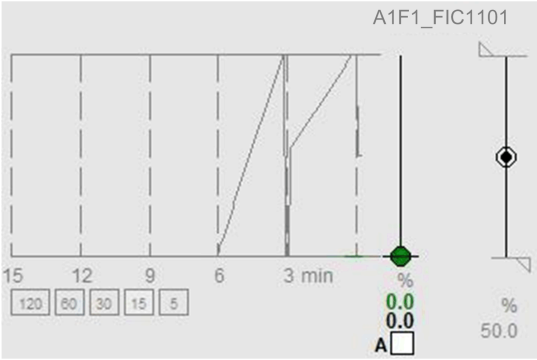
At the beginning of the manual, you can find a description of the icons, page 63 that are part of the genies.

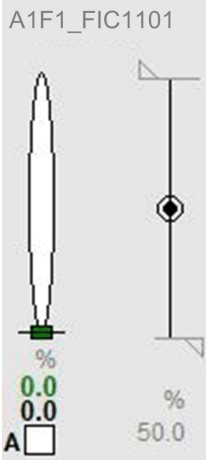
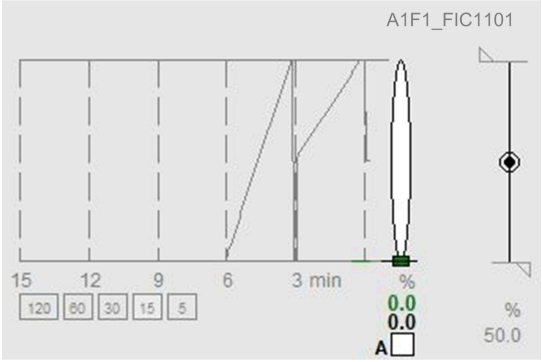
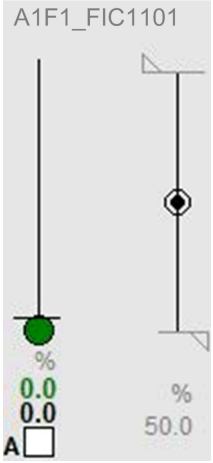
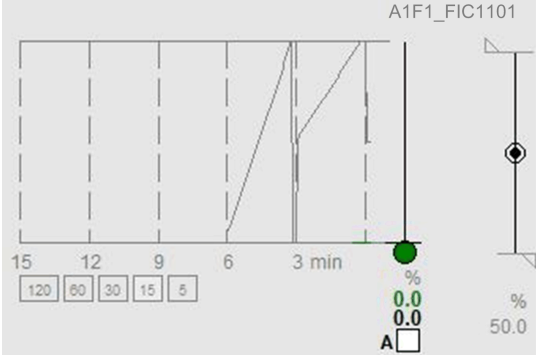
Representation

The table describes the genies of the gpl_pid library:

Graphic Symbol	Genie name	Description
	Bar_Op_Horiz	Horizontal bar with output display status.
	Bar_PVSP_Horiz	Horizontal bar with present value and setpoint display status.
	Display_PVSP	Present value and setpoint display status.
	Display_OP	Output display status.

Graphic Symbol	Genie name	Description
	Bar_PVSPOP_Horiz	Horizontal bar with present value, setpoint, and output display status.
	Display_PVSPOP	Present value, setpoint, and output display status.
	Bar_OP_Vert	Vertical bar with output display status.
	Bar_PVSP_Vert	Vertical bar with present value and setpoint, display status.
	Bar_PVSPOP_Vert	Vertical bar with present value, setpoint, and output display status.

Graphic Symbol	Genie name	Description
	Bar_PVSP_Vert_Trend	Vertical bar with present value and setpoint display status with trend.
	Bar_PVSPOP_Vert_Trend	Vertical bar with present value, setpoint, and setpoint display status with trend.
	Flow_PVSPOP_Vert	Displays the flow on a vertical bar with engineering units, present value (PV), setpoint (SP), and output (OP).
	Flow_PVSPOP_Trend	Displays the flow on a vertical bar with engineering units, present value (PV), and setpoint (SP), Output (OP). In addition, the symbol displays trends and allows you to select the trend period in minutes.

Graphic Symbol	Genie name	Description
	Pressure_PVSPOP_vert	Displays the pressure on a vertical bar with engineering units, present value (PV), setpoint (SP), and output (OP).
	Pressure_PVSPOP_Trend	Displays the pressure on a vertical bar with engineering units, present value (PV), setpoint (SP), and output (OP). In addition, the symbol displays trends and allows you to select the trend period in minutes.
	Temp_PVSPOP_Vert	Displays the temperature on a vertical bar with engineering units, present value (PV), setpoint (SP), and output (OP).
	Temp_PVSPOP_Trend	Displays the temperature on a vertical bar with engineering units, present value (PV), setpoint (SP), and output (OP). In addition, the symbol displays trends and allows you to select the trend period in minutes.

Faceplate

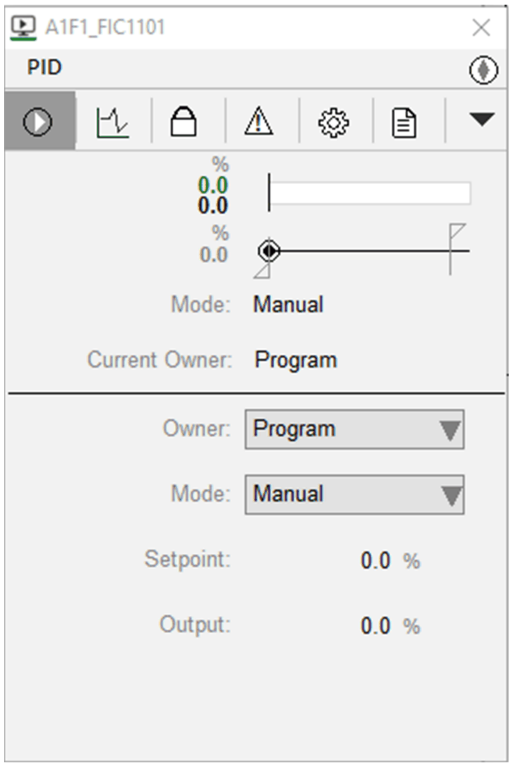
Overview

During operation, clicking a genie of the `gpl_pid` library allows you to display a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
  - Event, page 69
  - Trend, page 69
- Tabs for optional functions, which appear only if configured:
  - Interlock, page 87
  - Failure, page 78
  - Measures, page 70

Operator Tab

The figure shows an operator tab:



The table describes the control function of an operator tab:

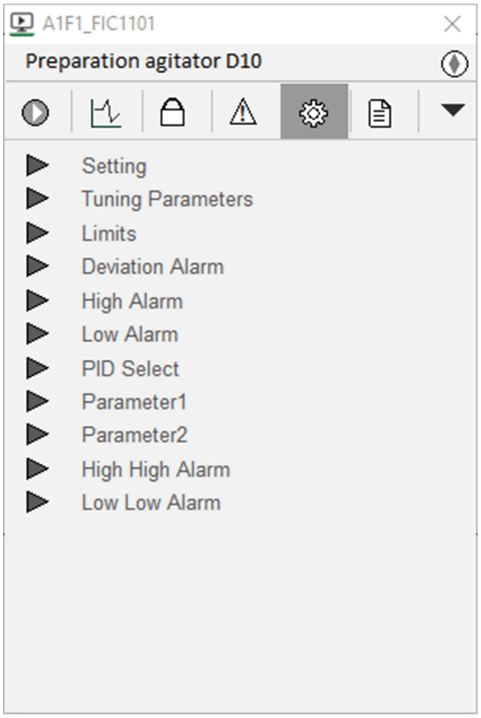
Label	Item name	Description	Security parameter, page 520
Mode	Mode	Current mode	–
Current Owner	Owner	Current owner of the `PID`.	Set owner
Owner	OwnerSelect	Owner of the `PID`. Refer to owner, page 58.	Set owner
Mode	Mode	Mode of operation.	–

Label	Item name	Description	Security parameter, page 520
Setpoint	Setpoint, OwnerSelect	Allows you to change the setpoint, if owner = operator.	Set setpoint
Output	Output, OwnerSelect	Output status of `PID`.	–

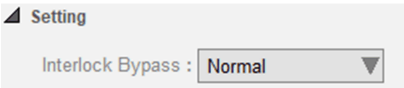
Engineering Tab

The table shows an engineering tab and its groups:

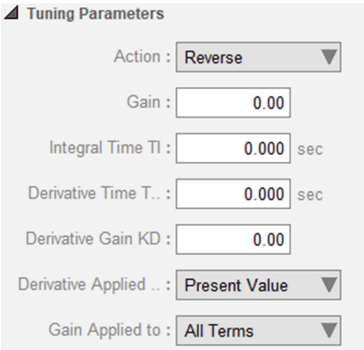
Engineering Tab



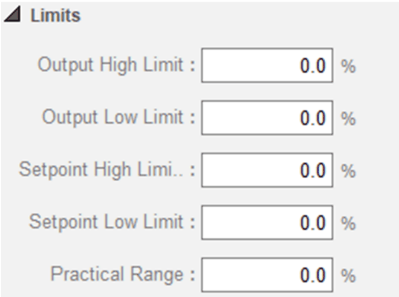
Setting



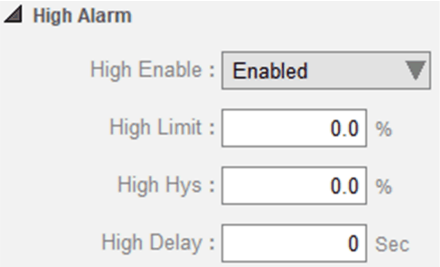
Tunning Parameters



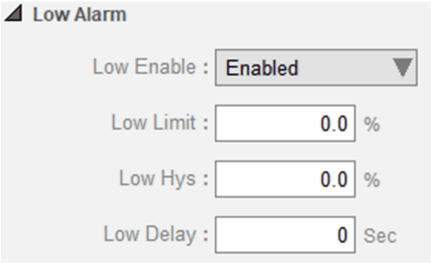
Limits



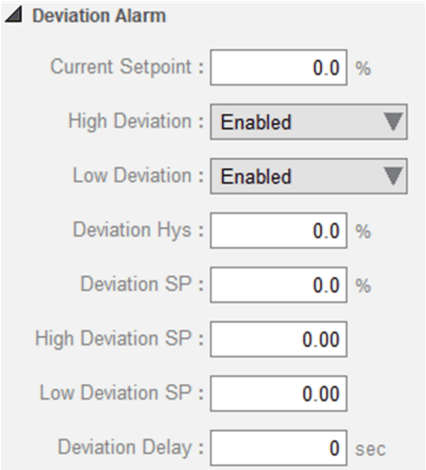
High Alarm



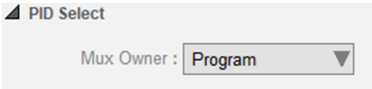
Low Alarm



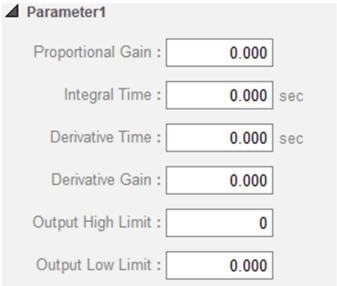
Deviation Alarm



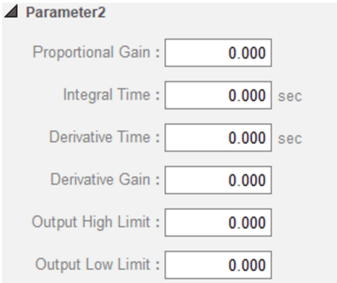
PID Select



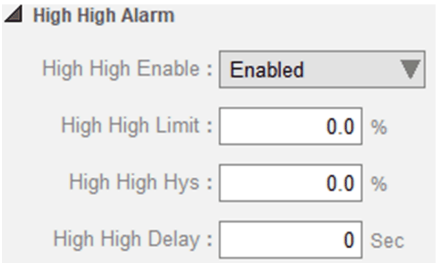
Parameter1



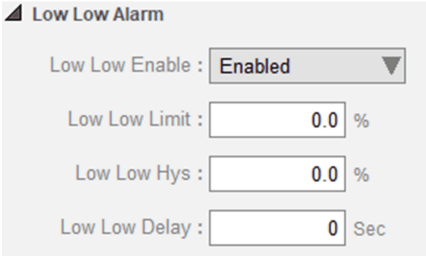
Parameter2



High High Alarm



Low Low Alarm



The table describes the control function of engineering tab:

Label	Item Name	Description	Security Parameter, page 520
Setting			
Interlock Bypass	InterlockBypass	Interlock bypass status Allows you to bypass interlock.	Set bypass conditions, set bypass all interlocks
Tuning Parameters			
Action	Action	Mode of PID operation	–
Gain	ProportionalGain	Proportional gain	Set parameters
Integral Time TI	IntegralTime	Integral time TD	Set parameters
Derivative Time TD	DerivativeTime	Derivative integral time TD	Set parameters
Derivative Gain KD	DerivativeGain	Derivative gain KD	Set parameters
Derivative Applied to	DerivativeAppliedTO	Output control action	–
Gain Applied to	MixPar	PID controller calculation formula is mixed or parallel	–
Limits			
Output High Limit	OutputHighLimit	Output high limit	Set parameters
Output Low Limit	OutputLowLimit	Output low limit	Set parameters

Label	Item Name	Description	Security Parameter, page 520
Setpoint High Limit	SetpointHighLimit	Setpoint high limit	Set parameters
Setpoint Low Limit	SetpointLowLimit	Setpoint low limit	Set setpoint
Practical Range	PR	Practical range	Set setpoint
Deviation Alarm			
Current Setpoint	Setpoint	Current setpoint	Set point
High Deviation	HighDeviationEnable	Enable high deviation	–
Low Deviation	LowDeviationEnable	Enable low deviation	–
Deviation Hys	DeviationHysteresis	Hysteresis value for deviation alarm.	Set parameters
Deviation SP	DeviationSetpoint	Maximum deviation setpoint	Set setpoint
High Deviation SP	DeviationHighSetpoint	High deviation setpoint	–
Low Deviation SP	DeviationLowSetpoint	Low deviation setpoint	–
Deviation Delay	DeviationDelay	Delay for enabling deviation	Set parameters
High Alarm			
High Enable	HighEnable	High alarm enable.	–
High Limit	HighLimit	High level setpoint.	Set setpoint
High Hys	HighLimitHysteresis	Hysteresis high.	Set parameters
High Delay	HighLimitDelay	Delay high.	Set parameters
Low Alarm			
Low Enable	LowEnable	Low alarm enable.	–
Low Limit	LowLimit	Low level setpoint.	Set setpoint
Low Hys	LowLimitHysteresis	Hysteresis low.	Set parameters
Low Delay	LowLimitDelay	Delay low.	Set parameters
PID Select			
Mux Owner	PIDMuxOwner	Owner of PID Mux.	Set owner
Parameter1			
Proportional Gain	PIDMuxProportionalGain1	Proportional gain GAIN1.	Set parameters
Integral Time	PIDMuxIntegralTime1	Integral time TI1.	
Derivative Time	PIDMuxDerivativeTime1	Derivative time TD1.	
Derivative Gain	PIDMuxDerivativeGain1	Derivative gain KD1.	
Output High Limit	PIDMuxOutputHighLimit1	Output high limit OPHILIM1.	
Output Low Limit	PIDMuxOutpuLowLimit1	Output low limit OPLOLIM1.	
NOTE: Parameter1 groups are configurable.			
Parameter2			
Proportional Gain	PIDMuxProportionalGain2	Proportional gain GAIN2.	Set parameters
Integral Time	PIDMuxIntegralTime2	Integral time TI2.	
Derivative Time	PIDMuxDerivativeTime2	Derivative time TD2.	

Label	Item Name	Description	Security Parameter, page 520
Derivative Gain	PIDMuxDerivative-Gain2	Derivative gain KD2.	
Output High Limit	PIDMuxOutputHigh-Limit2	Output high limit OPHILIM2.	
Output Low Limit	PIDMuxOutpuLowLi-mit2	Output low limit OPLOLIM2.	
NOTE: Parameter2 groups are configurable.			
High High Alarm			
High High Enable	HighHighEnable	High-high alarm enable.	–
High High Limit	HighHighLimit	High-high level setpoint.	Set setpoint
High High Hys	HighHighLimitHyste-resis	Hysteresis high-high.	Set parameters
High High Delay	HighHighLimitDelay	Delay high high.	Set parameters
Low Low Alarm			
Low Low Enable	LowLowEnable	Low-low alarm enable.	–
Low Low Limit	LowLow Limit	Low-low level setpoint.	Set setpoint
Low Low Hys	LowLowLimitHyste-resis	Hysteresis low-low.	Set parameters
Low Low Delay	LowLowLimitDelay	Delay low-low.	Set parameters

Items

Overview

This section describes the variables, equipstate variable, abnormal and action required conditions, alarms, events, and trends items of `PID`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`ProgramSetpointMode`	Setpoint source for program owner	Remote/Sequence	*Bit0*
`InterlockActive`	Interlock status	Interlocked/Healthy	*Bit2*
`OverrideOutput`	PID override output selection	OnOff	*Bit3*
`AutoConfigurationInvalid`	Auto mode configuration is invalid	–	*Bit4*
`Warning`	PID alert is active	–	*Bit5*
`OutputLow`	Output is less than or equal to the low threshold of output low limit	–	*Bit6*
`OutputHigh`	Output is more than or equal to the high threshold of output high limit	–	*Bit7*
`PVLimitIdentical`	Lower limit of the PV range is identical to the upper limit of the PV range	–	*Bit8*
Bit items derived from PIDTuneStatusWord.			
`PIDTuneCalculationError`	PID tuning error is detected in a floating point value calculation	–	*Bit0*

Item name	Description	Enumeration	Address
PIDTuneInvalidInput	Invalid value recorded at one of the floating point inputs		*Bit1*
PIDTuneDivisionByZero	Division by zero with calculation in floating point values		*Bit2*
PIDTuneCapacityOverflow	Capacity overflow during floating point value calculation		*Bit3*
PIDTunePerfOutOfRange	Parameter perf is outside the range		*Bit4*
PIDTuneThreshold	Thresholds pv_inf and pv_sup of the controller to be set		*Bit7*
PIDTuneParaCConnected	PARA_C output is not connected to the parameters of an autotunable controller		*Bit8*
PIDTuneFailed	Autotuning unsuccessful		*Bit9*
PIDTuneSuccess	Last autotune was successful		*Bit10*
Bit items derived from PIDTuneDiagnosisWord.			
PIDTuneRunning	Autotuning is running	Active/Inactive	*Bit0*
PIDTuneAborted	Autotuning aborted		*Bit1*
PIDTuneParError	Parameter error		*Bit2*
PIDTuneAlterPar	Alteration of parameters which have just been set automatically		*Bit3*
PIDTuneSysError	Stop as a consequence of detected system error		*Bit4*
PIDTuneSaturated	Process value saturated		*Bit5*
PIDTuneAlterSmall	Alteration too small		*Bit6*
PIDTuneSamplInvalid	Sampling interval invalid		*Bit7*
PIDTuneIncomReact	Incomprehensible reaction		*Bit8*
PIDTuneNonStabMeas	Non-stabilized measuring at the start		*Bit9*
PIDTuneTMaxShort	Length of actuating pulse Stellimpulses too short		*Bit10*
PIDTuneNoise	Too much noise/interference		*Bit11*
PIDTuneTMaxLong	Length of actuating pulse Stellimpulses too long		*Bit12*
PIDTuneExThresh	Process with significant exceeding of the thresholds		*Bit13*
PIDTuneMinPhase	Process without minimum phase		*Bit14*
PIDTuneAsymProcess	Asymmetric process		*Bit15*
Bit items derived from PIDTuneConfigurationWord.			
PIDTuneStart	Start autotune command	–	*Bit0*
PIDTuneStop	Stop autotune command		*Bit1*
PIDTuneTransfer	Transfer parameters		*Bit2*
PIDTuneRetPrevPar	Return previous parameters		*Bit3*
Bit items derived from PIDMuxConfigurationWord2.			
PIDMuxAction2	Mode of PID operation	–	*Bit3*
PIDMuxDerivativeApplied-To2	Output control action		*Bit4*
PIDMuxMixPar2	PID controller calculation formula is mixed or parallel		*Bit5*
Bit items derived from PIDMuxConfigurationWord1.			
PIDMuxAction1	Mode of PID operation	–	*Bit3*
PIDMuxDerivativeApplied-To1	Output control action		*Bit4*

Item name	Description	Enumeration	Address
PIDMuxMixPar1	PID controller calculation formula is mixed or parallel		*Bit5*
Bit items derived from PIDMuxConfigurationWord.			
PIDMuxOwnerSelect	Owner of control module	–	*Bit0*
PIDMuxParaSelect	Owner of control module		*Bit1*
Bit items derived from ConfigurationWord.			
OwnerSelect	Owner of control module	Operator/Program	*Bit0*
InterlockBypass	Interlock bypass command	Bypassed/Normal	*Bit1*
Mode	PID operating mode selection	Auto/Manual	*Bit2*
Action	Mode of PID operation	Direct/Reverse	*Bit3*
DerivativeAppliedTo	Output control action	Error/Present Value	*Bit4*
MixPar	PID controller calculation formula is mixed or parallel	Only Proportional/All Terms	*Bit5*
IO Items			
StatusWord	Status word	–	PIDCTLGP_ST.STW
ConfigurationWord	Configuration word		PIDCTLGP_ST.CFGW
PIDTuneDerTime	Derivative time	–	PID_PIDTuneDerTime
PIDTuneIntTime	Integral time	–	PID_PIDTuneIntTime
PIDTuneGain	Proportional gain	–	PID_PIDTuneGain
PIDTuneConfigurationWord	Configuration word bit array	–	PID_PIDTUNEGP_ST_CFGW
PIDTuneDiagnosisWord	Diagnosis word bit array	–	PID_PIDTUNEGP_ST_ DIAGW
PIDTuneStatusWord	Status word bit array	–	PID_PIDTUNE_ST_STW
SetpointLowLimit	Define the minimum value of the setpoint	–	PID_SetpointLowLimit
SetpointHighLimit	Define the maximum value of the setpoint	–	PID_SetpointHighLimit
ActionRequired	Action required	–	PID_ActionRequired
Abnormal	Abnormal	–	PID_Abnormal
DerivativeGain	Derivative gain	–	PID_DerivativeGain
DerivativeTime	Derivative time	–	PID_DerivativeTime
IntegralTime	Integral time	–	PID_IntegralTime
ProportionalGain	Proportional gain	–	PID_Gain
OutputLowLimit	Define the minimum value of the output	–	PID_OutputLowLimit
OutputHighLimit	Define the maximum value of the output	–	PID_OutputHighLimit
Output	Output	–	PID_Output
Setpoint	Setpoint	–	PID_Setpoint
AutoSP	Setpoint	–	PID_AutoSP
SPIncRate	Setpoint increment rate	–	PID_SPIncRate
SPDecRate	Setpoint decrement rate	–	PID_SPDecRate
OutRate	Output rate	–	PID_OutRate
PresentValue	Present value of measurement	–	PID_PresentValue
PIDTunePrevDerTime	Autotune previous autotune derivative time	–	PID_ PIDTunePrevDerTime
PIDTunePrevIntTime	Autotune previous integration time	–	PID_ PIDTunePrevIntTime

Item name	Description	Enumeration	Address
PIDTunePrevGain	Autotune previous proportional constant	–	PID_PIDTunePrevGain
PIDTuneTime	Autotune time	–	PID_PIDTuneTuneTime
PIDTuneAmp	Step amplitude	–	PID_PIDTuneTuneAmp
PIDMuxOutputHighLimit1	Define the maximum value of the output	–	PIDMultiplexer_OutputHighLimit1
PIDMuxOutputLowLimit1	Define the minimum value of the output	–	PIDMultiplexer_OutputLowLimit1
PIDMuxProportionalGain1	Proportional gain	–	PIDMultiplexer_Gain1
PIDMuxIntegralTime1	Integral time	–	PIDMultiplexer_IntegralTime1
PIDMuxDerivativeTime1	Derivative time	–	PIDMultiplexer_DerivativeTime1
PIDMuxOutputHighLimit2	Define the maximum value of the output	–	PIDMultiplexer_OutputHighLimit2
PIDMuxOutputLowLimit2	Define the minimum value of the output	–	PIDMultiplexer_OutputLowLimit2
PIDMuxProportionalGain2	Proportional gain	–	PIDMultiplexer_Gain2
PIDMuxIntegralTime2	Integral time	–	PIDMultiplexer_IntegralTime2
PIDMuxDerivativeTime2	Derivative time	–	PIDMultiplexer_DerivativeTime2
PIDMuxConfigurationWord	Multiplexer configuration word bit array	–	PID_PIDMUX_ST_CFGW
PIDMuxConfigurationWord1	Multiplexer configuration word bit array	–	PID_PIDMUX_ST_CFGW1
PIDMuxConfigurationWord2	Multiplexer configuration word bit array	–	PID_PIDMUX_ST_CFGW2
PID_PIDMUX_ST_STW	Multiplexer status word bit array	–	PID_PIDMUX_ST_STW
PIDMuxPIDSelect	Parameter set selection	–	PID_PIDMuxPIDSelect
ORDsp	Optimal range display	–	TRUE
ORHigh	Optimal high	–	PID_PIDGP_ORHigh
ORLow	Optimal low	–	PID_PIDGP_ORLow
PR	Practical range	–	PID_PIDGP_PR
PRHigh	Practical range high	–	PID_PIDGP_PRHigh
PRLow	Practical range low	–	PID_PIDGP_PRLow
OptimalDevHigh	Optimal deviation high	–	PID_PIDGP_OptimalDevHigh
OptimalDevLow	Optimal deviation low	–	PID_PIDGP_OptimalDevLow
Calculated Items			
EquipState	Output information For more details, refer to EquipState calculated variable, page 548.	–	GPL_PIDEquipState
Owner	Current owner of the PID For more details, refer to calculated variable, page 58.		GPL_Owner
ModeSymbol	Current mode of the PID For more details, refer to calculated variable, page 59.		GPL_PIDMode

Item name	Description	Enumeration	Address
Abnormal	Shows abnormal condition of the PID For more details, refer to abnormal conditions, page 548.		GPL_PIDAbnormal
ActionRequired	Shows action required condition of the PID For more details, refer to action required conditions, page 548.		GPL_PIDActionRequired

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	Conditions	Enumeration
Active with Interlock	ILCK x OP	On Interlocked
Passive with Interlock	ILCK x Not OP	Off Interlocked
Active	Not ILCK x OP	On
Passive	Not ILCK x Not OP	Off

Abnormal Conditions

The table describes the abnormal conditions of PID:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
AbnormalOwner	Abnormal owner status appears if the current owner is not a normal owner which configured at the instance level.	*Bit0*
AbnormalMode	Abnormal mode status appears if the current operating mode is not normal mode.	*Bit1*
AbnormalInterlockBy-pass	Abnormal interlock bypass condition appears if the Interlock bypass is enabled.	*Bit2*
AbnormalAction	Abnormal action	*Bit3*
AbnormalHighDeviatio-nEnable	If low alarm enable is active.	*Bit4*
AbnormalLowLowEnable	If low-low alarm enable is active.	*Bit5*
AbnormalLowEnable	If low alarm enable is active.	*Bit6*
AbnormalLowDeviatio-nEnable	If low alarm enable is active.	*Bit7*
AbnormalOperatorTab	Abnormal status of operator tab. This condition appears if the Interlock bypass is enabled.	*Bit8*
AbnormalHighHighEnable	If high-high alarm enable is active.	*Bit9*
AbnormalHighEnable	If high alarm enable is active.	*Bit10*
AbnormalInterlockTab	Abnormal status of interlock tab.	*Bit11*
AbnormalEngineeringTab	Abnormal status of engineering tab.	*Bit12*

Action Required Conditions

The table describes the action required conditions of PID:

Item name	Description	Address
Bit items derived from ActionRequired variable word.		
`ActionInterlockTab`	Action required status of interlock tab.	*Bit11*

Alarms

Alternate Services:

- LAlarm, page 123
- DAlarm, page 128

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
`StateAlarm`	All changes in the PID status and output.
`ModeAlarm`	All changes in the mode of operation of the equipment.
`ActionAlarm`	All the operator actions in the supervision.

Trends

The table describes the trend tags that are managed by Supervision components and their expression:

Item name	Description	Expression
`PV`	Present value	`PIDCTLGP_ST.PV`
`Setpoit`	Current setpoint	`PIDCTLGP_ST.SP`
`OP`	Current output	`PIDCTLGP_ST.OP`
`HighHighLimit`	High-high level set Point	`LALARMGP_CFG.SPHH`
`LowLowLimit`	Low-low level set point	`PIDCTL_CFG.SPLL`
`LowLimit`	Low level set point	`PIDCTL_CFG.SPL`
`HighLimit`	High level set Point	`PIDCTL_CFG.SPH`
`DeviationSetpoint`	Maximum deviation setpoint	`PIDCTL_CFG.SPD`
`ORHigh`	Optimal range high	`PIDCTL_CFG.ORH`
`ORLow`	Optimal range low	`PIDCTL_CFG.ORL`

NOTE: `ORHigh` and `ORLow` are disabled when optimal range parameter is disabled.

Ratio Controller Management

What's in This Chapter

Template 550
Control – RATIOCTLGP 554
Control – ADVCTLDATAR 557
Supervision 558

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the management of ratio controller.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$RatioCtrlGP` template.

Description

General Description

The `$RatioCtrlGP` process control module template is used to condition the signals associated with a standard Control library RATIO controller.

It helps monitor and control the controller from the Supervision system and provides the operating modes used in the remaining EcoStruxure Process Expert process templates of one stream in a process at a defined or specified proportion relative to that of another.

The RATIO controller sets a setpoint that is based on a measurement signal and on the ratio that needs to be maintained between the measurement signal and the setpoint. This process template is especially used to combine or blend two feed streams to produce a mixed flow with a desired composition or physical property.

Function Description

The main functions of the ratio control template are described in the following table:

Function	Description
Ratio	The template includes and incorporates functions provided by the RATIO controller from the standard `CONT_CTL` Control library.
Owner	The object manages control system level, which is the owner (Operator or Program). As a result, it is responsible for setting the set-point and activation for the controller.

Function	Description
Local and remote set-point	Enables to control the template with a set-point that is determined through a set-point selector (local or remote) in program owner. The local set-point is assigned from the sequential control while the remote set-point is assigned to the template from the logic implemented in the continuous control.
Downstream object tracking	When `$RatioCtrlGP` template is connected with downstream objects as Advanced Controllers (`$PIDADVGP` template of General Purpose Library or `$PSFAP/$PIFAP/$MSFAP/$MIFAP` templates of APC SA Library). If these downstream objects are either in Operator Owner or in Program-Local or in Manual Mode, then `$RatioCtlGP` output shall be the Process value of the downstream objects. Else, `RatioCtrlGP` output shall be as per this function block's logic/formula.

Parameters

Parameters

The `$RatioCtrlGP` template defines operation of ratio control. Alternative ratio control facets enable the modification of the ratio control that are available.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Oprange	Output high range (EU)	`Float`	100.0	High range of output from ratio control.
	Output low range (EU)		0.0	Low range of output from ratio control.
	Output engineering units	`String`	%	Engineering units of output from ratio control.
	Output format type		####.#EU	Format type of output from ratio control.
Krange	K high range (EU)	`Float`	100.0	High range of Ratio coeffiecient.
	K low range (EU)		0.0	Low range of Ratio coeffiecient.
	K engineering units	`String`	%	Engineering of Ratio coeffiecient.
	K format type		####.#EU	Format type of Ratio coeffiecient.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Operation	Normal owner	`Enum`	Program and Cascade (5)	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	Set owner	`Enum`	Operator (1)	Privilege to change owner. This parameter is used in Operator tab, page 560.

Section	Parameter description	Type	Default	Additional remarks
	Set setpoint			Privilege to change setpoint. This parameter is used in Operator tab, page 560.
Trend	Sample period	Duration	00:00:05	Defines the rate at which the trend data is displayed on the trend page Example: Log Data after 5 seconds if sample period 00:00:05.
	History length (weeks)	Integer	5	The duration of historical data Example: Historized Data of 5 weeks in one file with a configured sampling period.
	History rollover time	Duration	12:00:00	The time at which the new file generates for historising based on the rollover day configuration.
	History rollover day	String	Tuesday	The day at which the new file generates for historising.
	Storage type		TRN_EVENT	A trend that is sampled continuously at a specified period.
	Data location (optional)		[Data]:	User configurable file location.
Configuration	Format of offset coefficient	String	#####.###	–
Deadband	Deadband of output	Integer	0	–
	Deadband of present value		0	–
	Deadband of present value tracking		0	–
	Deadband of present ratio coefficient		0	–
	Deadband of ratio offset coefficient		0	–
	Deadband of offset coefficient		0	–
Events	Enable log	Boolean	False	Enable event logging for asset status.
Historize	Hstorize status/ commands	Boolean	False	It allows to historize the change status/ commands.
	Historize output			–
	Historize present value			It allows to historize the change present value.
	Historize present value tracking			–
	Historize present ratio coefficient			–
	Historize ratio offset coefficient			–
	Historize offset coefficient			–
Trend	Disable output trend	Boolean	False	–

Section	Parameter description	Type	Default	Additional remarks
	Disable present value tracking trend			–

Display Parameters

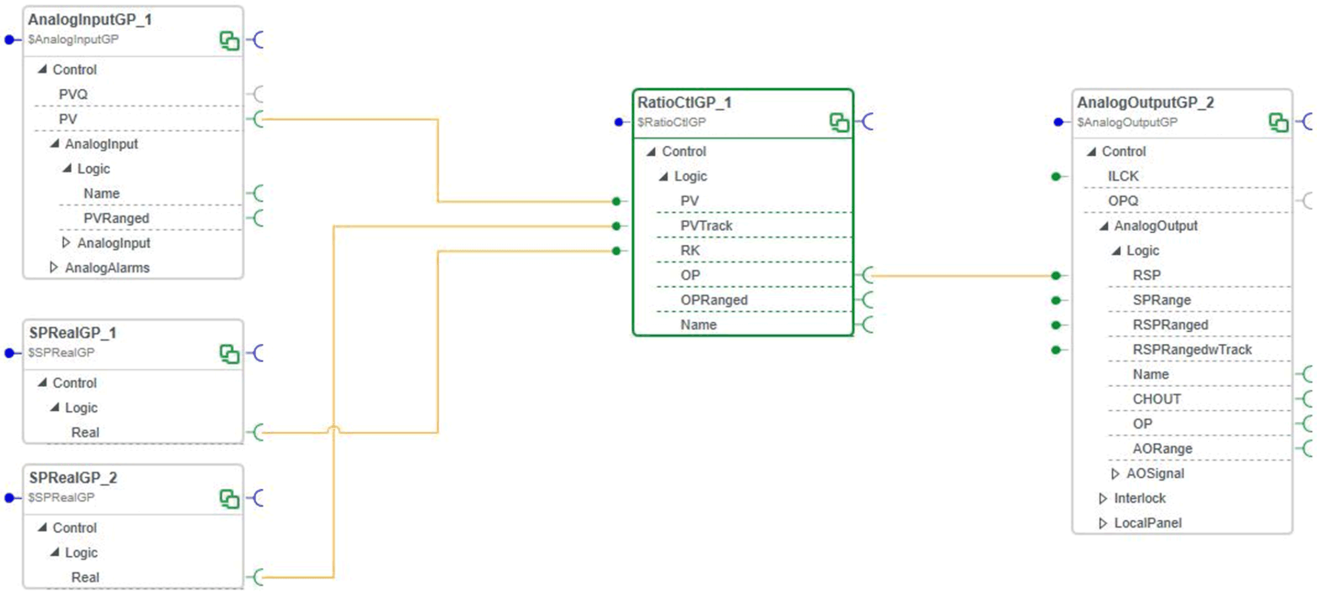
For more details, refer to the topic describing the parameters of display, page 43.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$RatioCtrlGP` (also applicable for `$RatioCtrlCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



The following table lists the link interface that creates link symbol on faceplate:

Interface name	Runtime parameter
`RK`	LKNavigate
`PV`	PVNavigate
`OP`	OPNavigate
`OPRanged`	
`PVTrack`	PV_TRACKNavigate

For more details, refer to Supervision chapter > Faceplate > Navigation Link of General Purpose Library User Guide.

Refer to common control module template interface, page 47 for more details on common interfaces.

The table describes the control module template interfaces:

Interface identifier	Type/Role	Description	Example
`PVTrack`	*$Real/Ref*	Reference variable of control loop input interface from other template.	A real value can be connected as reference to maintain one ratio between PV and PV track

Interface identifier	Type/Role	Description	Example
RK	*$Real/Ref*	Remote relationship coefficient input interface from other template.	
OP	*$Real/Def*	Current output interface to other template.	

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control – RATIOCTLGP

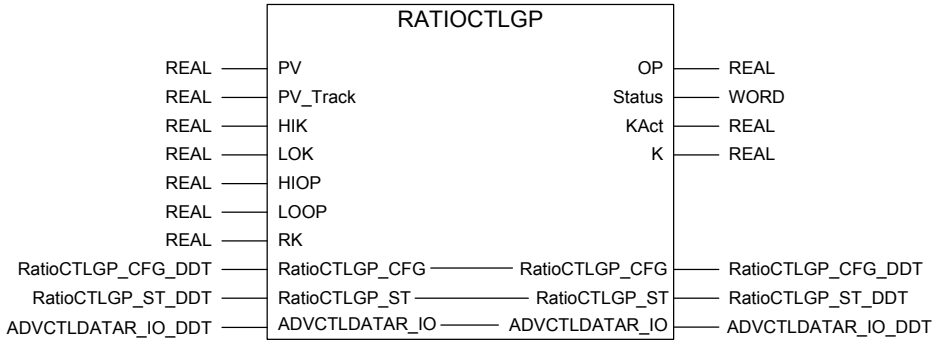
Overview

This chapter describes the RATIOCTLGP DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
PV	REAL	Current value of the variable that is being controlled by the controller.
PV_Track	REAL	Current value of the reference variable.
HIK	REAL	Upper value of the ratio that needs to be maintained between PV and PV_Track. Used to limit the maximum constant admitted by the controller. This input is internally connected to the PARA.K_MAX parameter of the RATIO controller.
LOK	REAL	Lower value of the ratio that needs to be maintained between PV and PV_Track. Used to limit the minimum constant admitted by the controller. This input is internally connected to the PARA.K_MIN parameter of the RATIO controller.
HIOP	REAL	Upper range of the output (OP) calculated by the RATIO controller. Used to limit the output calculated by the controller. This input is internally connected to the PARA.SP_MAX parameter of the RATIO controller.

Parameter	Type	Description
LOOP	REAL	Lower range of the output (OP) calculated by the RATIO controller. Used to limit the output calculated by the controller. This input is internally connected to the PARA.SP_MIN parameter of the RATIO controller.
RK	REAL	Remote ratio set-point. Enables to define the desired ratio from the continuous control. Refer to the SC.Rem public variable, page 556.

Outputs

Output Parameter Description

Parameter	Type	Description
OP	REAL	Output generated by the RATIO controller based on the measurement and the configured ratio. OP is calculated as: K x PV_Track + Rem_CFG.Bias.
Status	WORD	Statuses output by the RATIO controller on the Status output. Refer to the RATIO controller in Control documentation for more details.
KAct	REAL	Actual ratio coefficient generated by the RATIO controller. KAct is calculated as: (PV – Rem_CFG.Bias) / PV_Track.
K	REAL	Ratio coefficient for the ratio between the generated output and the measurement that is being applied on the basis of the operating mode.

The DFB performs the ratio coefficient calculation based on the signal statuses. The ratio coefficient calculation is detailed in the following table:

Owner	Rem	KD is calculated as:
OFF	OFF	SC.LK
OFF	ON	RK
ON	-	Rem_ST.LK You can modify the ratio coefficient directly from the monitoring subsystem while this mode is active.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
RatioCTLGP_CFG	RatioCTLGP_CFG_DDT	Provides the data needed to configure the DFB normally from the monitoring subsystem.
RatioCTLGP_ST	RatioCTLGP_ST_DDT	Provides the data needed to monitor and/or control the DFB status.
ADVCTLDATAR_IO_DDT	ADVCTLDATAR_IO_DDT	Advance control data structure.

RatioCTLGP_CFG_DDT Type

Name	Type	Description
Bias	REAL	Enables to define the bias parameter for the RATIO controller. Used for calculating the OP output.
KAct	REAL	Read-only access Refer to the KAct output pin, page 555.

RatioCTLGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	Rem	Bit0	BOOL	Refer to the Rem, page 556.
CFGW			WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
	Owner	Bit0	BOOL	Read/write access Enables to configure whether the ratio set-point is set by the Program (0) or the Operator (1).
PV	–	–	REAL	Read-only access Refer to the PV input pin, page 554.
PV_Track	–	–	REAL	Read-only access Refer to the PV_Track input pin, page 554.
OP	–	–	REAL	Read-only access Refer to the OP output pin, page 555.
LK	–	–	REAL	Read/write access Enables to set the ratio when the owner is the Operator (Owner = 1).

ADVCTLDATAR_IO_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	OPTrack-En	*Bit2*	BOOL	1 = output tracking enable.
TrackValue	–	–	REAL	Track value

Public Variables

Public Variable Description

Variable	Type	Description
SC	RatioCTLGP_SC_DDT	Provides the frequently needed data to monitor and control the controller status from the sequential control.

RatioCTLGP_SC_DDT Type

Name	Type	Description
OP	REAL	Read-only access Refer to the OP output pin, page 555.
PV_Track	REAL	Read-only access Refer to the PV_Track input pin, page 554.
PV	REAL	Read-only access Refer to the PV input pin, page 554.
Owner	BOOL	Read-only access Refer to the Rem_ST.CFGW.Owner input/output pin, page 555.
Rem	BOOL	Read-only access 1 = Allows to configure the DFB for a remote set-point -RSP, if the owner is the Program 0 = Allows to configure the DFB for a local set-point -LSP, if the owner is the Program.
LK	REAL	Read/write access Enables the sequential control to assign the local ratio set-point if the owner is the Program (Owner is 0) and the selected set-point is local (SC.Rem is 0). Otherwise, the current set-point (K output) is continuously copied to this variable.

Control – ADVCTLDATAR

Overview

This chapter describes the ADVCTLDATAR DFB.

Description

This ADVCTLDATAR DFB is generated when $RatioCtrlGP template is connected with downstream objects as Advanced Controllers (example, $PIDADVGP template of General Purpose Library or $PSFAP/$PIFAP/$MSFAP/$MIFAP templates of APC SA Library).

It takes the downstream object's information (e.g. Owner status, Remote status, mode status and Process Value) to decide when Output Tracking shall be enabled.

Output Tracking shall be enabled in the below mentioned conditions:

- DownStream object owner is Operator.
- DownStream object is in Program-Local (i.e Owner =0 & REM = 0).
- DownStream object is in Manual Mode.

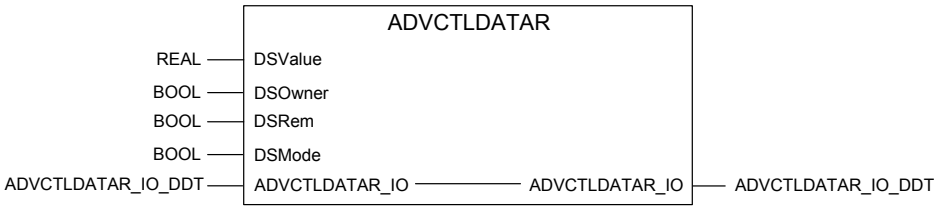
Output Tracking enable information and DownStream value (Controller's Process Value) is passed to this Input-output variable.

$RatioCtrlGP Template generates the same variable on ADVCTLDATAR_IO pin of both RatioCtrlGP and ADVCTLDATAR DFBs, So the downstream object information gets available in the RatioCtrlGP DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
DSValue	REAL	Downstream DFB output value.
DSOwner	BOOL	Owner (0 = Owner, 1 = Operator).
DSRem	BOOL	1 = Remote mode selected.
DSMode	BOOL	1 = Auto, 0 = Manual.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
ADVCTLDATAR_IO	ADVCTLDATAR_IO_DDT	Advance control data structure.

ADVCTLDATAR_IO_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	OPTrack-En	*Bit2*	BOOL	1 = output tracking enable.
TrackValue	–	–	REAL	Track value

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the management of ratio controller.

Genies

Genie Properties

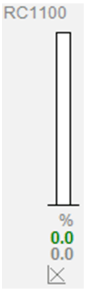
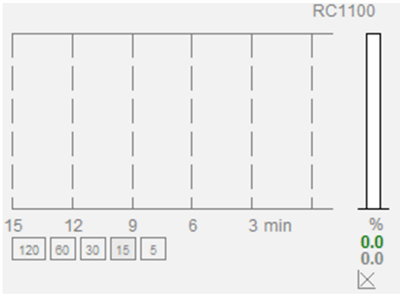
Refer to Genie Properties, page 62

Genie Icons

Refer to Genie Icons, page 63

Representation

The table describes the genies of the `gpl_ratiocontrol` library:

Graphic symbol	Genie name	Description
	Bar_PVOP_Horiz	Displays the label on a horizontal bar with engineering units, present value (PV), and output (OP).
	*Bar_PVOP_Vert*	Displays the label on a vertical bar with engineering units, present value (PV), and output (OP).
	*Bar_PVOP_Vert_Trend*	Displays the label on a vertical bar with engineering units, present value (PV), and output (OP). In addition, the symbol displays trends and allows you to select the trend period in minutes.
	Display_PVOP	Displays the label with engineering units, present value (PV), and output (OP).

Faceplates

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

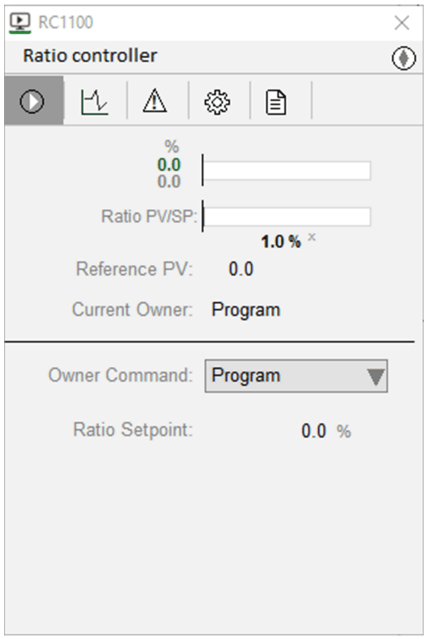
Available Tabs

During operation, clicking an `ratio control` genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
- Tabs for optional functions, which appear only if configured:
  - Interlock, page 87
  - Failure, page 78

Operator Tab

The figure shows an operator tab:

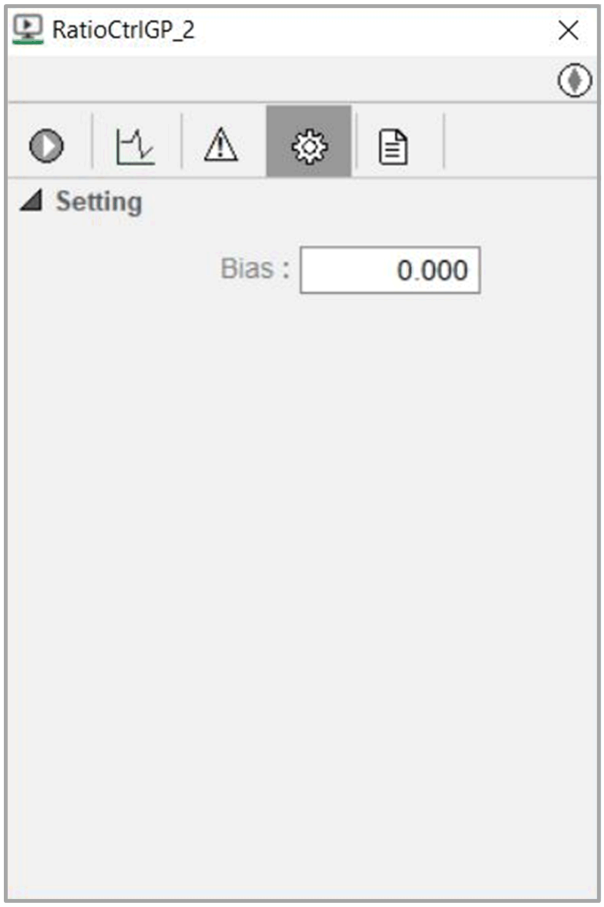


The table describes the control function of an operator tab:

Label	Item name	Description	Security parameter, page 551
–	PV, SP	Present value and setpoint	–
Ratio PV/SP	KACT, LK	Present ratio coefficient	
Reference PV	PV_TRACK	Reference value of the control loop	
Current Owner	Owner	Current owner of ratio control. For more details, refer to operating owner, page 58.	Set owner
Owner Command	OwnerCommand	Current owner of ratio control. For more details, refer to operating owner, page 58.	
Ratio Setpoint	LK, OwnerSelect	Local coefficient ratio	Set setpoint

Engineering Tab

The figure shows an engineering tab:



The table describes the control function of engineering tab:

Label	Item name	Description	Security parameter, page 551
Bias	Bias	Offset coefficient	set setpoint

Items

Overview

This section describes the variables, abnormal and action required conditions, alarm items, events, and trend item of `ratio control`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`ProgramSetpointMode`	Setpoint source for program owner	Active/Inactive	*Bit0*
Bit items derived from ConfigurationWord.			
`OwnerSelect`	Owner of control module	Operator/Program	*Bit0*
IO Items			

Item name	Description	Enumeration	Address
StatusWord	Status word	–	RATIOCTLGP_ST.STW
ConfigurationWord	Configuration word		RATIOCTLGP_ST.CFGW
Calculated Items			
Owner	Current owner of the ratio control For more details, refer to calculated variable, page 58.	–	GPL_Owner
Abnormal	Shows abnormal condition of the ratio control For more details, refer to Abnormal Conditions, page 562.		GPL_RatioCtlAbnormal

Abnormal Conditions

The table describes the abnormal conditions of ratio control:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
AbnormalOwner	Abnormal owner status appears if the current owner is not a normal owner which configured at the instance level.	*Bit0*
AbnormalOperatorTab	Abnormal status of operator tab. This condition appears if the Interlock bypass is enabled.	*Bit8*
AbnormalInterlockTab	Abnormal status of interlock tab.	*Bit11*

Action Required Conditions

The table describes the action required conditions of ratio control:

Item name	Description	Address
Bit items derived from ActionRequired variable word.		
RearmRequired	Action required status for rearm condition.	*Bit0*
ActionOperatorTab	Action required status of operator tab.	*Bit8*
ActionFailTab	Action required status of failure tab.	*Bit10*
ActionInterlockTab	Action required status of interlock tab.	*Bit11*

Alarms

The table describes the advanced alarm items that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expressions
Channel-Failure	Operations counter overflow	GPL_BitValueCheck (StatusWord, 1)

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
`ActionAlarm`	All actions by the operator.

Trends

The table describes the trend items that are used by Supervision components:

Item name	Expression	Address
`PV`	Present value	`RATIOCTLGP_ST.PV`
`OP`	Current output	`RATIOCTLGP_ST.OP`

Ramp

What's in This Chapter

Template 564
Control 567
Supervision 571

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the ramp.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$RampGP` template.

Description

General

The `$RampGP` process control module template is used to generate rising or falling setpoint ramps for other control blocks, such as PID regulators, direct analog outputs:

The template provides various functions, for example:

- Tracking (for example, for setpoints or measurements).
- Temporary ramp shutdown due to maximum deviation.
- Management of setpoints for monitoring and/or control.

Function Description

The main functions of ramp template are described in the following table:

Function	Description
Ramp	The template allows to generate raising and/or falling ramps with different slopes as required.
Maximum Deviation	Optionally, configure the template so that the ramp is held until the deviation between the measurement and the setpoint is less than the maximum deviation allowed.
Tracking	While the $RampGP template is not in operation, the signal on one of its pins is tracked (normally the pin connected to the measurement that needs to be controlled or to the last position of the control element – for example, a control valve).
Owner	The object manages control system level, which is the owner (Operator or Program). Therefore, it is responsible for setting the ramp setpoints.
Local/Remote Setpoint	If the Program is the owner, you can set the ramp setpoint from continuous control (with the input pin designated for this purpose) or from the sequence (with the `SC` public variable).

Parameters

Parameters

The `$RampGP` template provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Setpoint	Force remote setpoint	`Boolean`	False	Enables the remote control operation for ramp.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Operation	Normal owner	`Enum`	Program & Cascade (5)	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	Set owner	`Enum`	Operator (1)	Privilege to change owner. This parameter is used in Operator Tab, page 572.
	Set setpoint		Operator (1)	Privilege to change setpoint. This parameter is used in Operator Tab, page 572.
	Set parameters		Engineer (5)	Privilege to set parameters. This parameter is used in Engineering Tab, page 572.
Asset and Display	For more details, refer to the topic describing the parameters of asset and display, page 43.			
Trend	For more details, refer to the topic describing the parameters of trend, page 47.			

Data

The table describes the data parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Gradients format	`String`	#####.#EU	Format for raising gradient, falling gradient, and maximum deviation.
	Gradients maximum value	`Float`	100.0	Maximum value of raising gradient and falling gradient
Deadband	Setpoint	`Integer`	0	Deadband value for setpoint.
	Target setpoint			Deadband value for target setpoint.
Events	Enable log	`Boolean`	False	It allows to historize the operator actions.
Trend	Disable setpoint	`Boolean`	False	It disables setpoint trend tag.

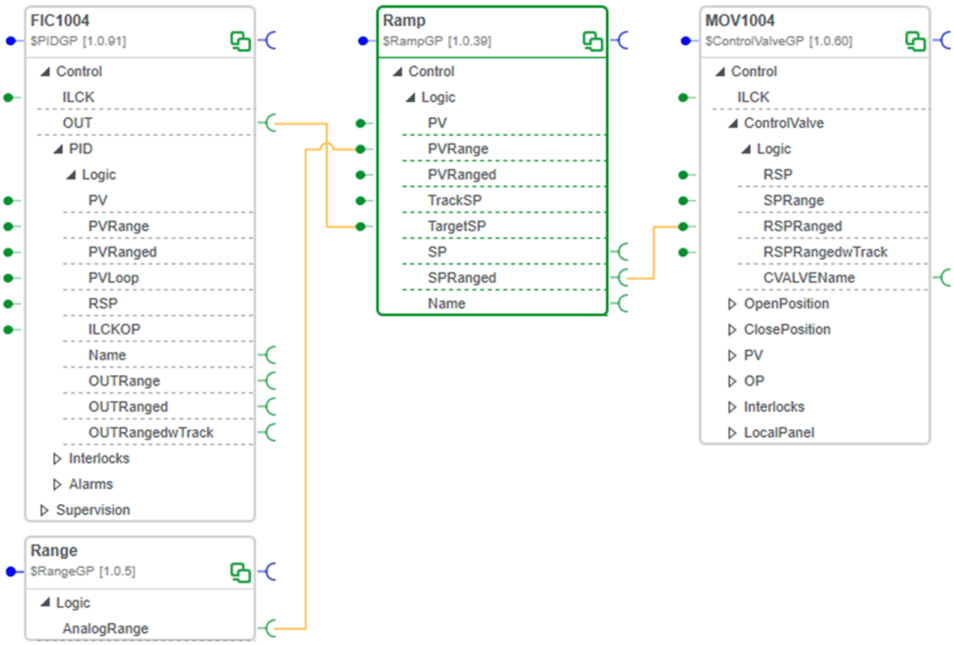
Section	Parameter description	Type	Default	Additional remarks
	Disable target setpoint			It disables target setpoint trend tag.
Historize	Gradients and maximum deviation	`Boolean`	False	It allows to historize the raising gradient, falling gradient, and maximum deviation.
	Setpoint			It allows to historize the change in setpoint.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$RampGP` (also applicable for `$RampCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



The following link interface create the link symbol on faceplate:

- SP or SPRanged
- TargetSP

For more details, refer to Supervision chapter > Faceplate > Navigation Link of General Purpose Library User Guide.

The table describes the control module template interfaces:

Interface identifier	Type/Role	Description	Example
`TrackSP`	*$Real/Ref*	Track SP value	Track setpoint while ramp is disable.
`TargetSP`	*$Real/Ref*	Target setpoint	Setpoint value is ramped and output is fed to the control element.
`Name`	*$ARAMPIFGP/Def*	Instance name concatenated with control block name	–

Refer to, page 47 common control module template interface for more details on common interfaces.

Control

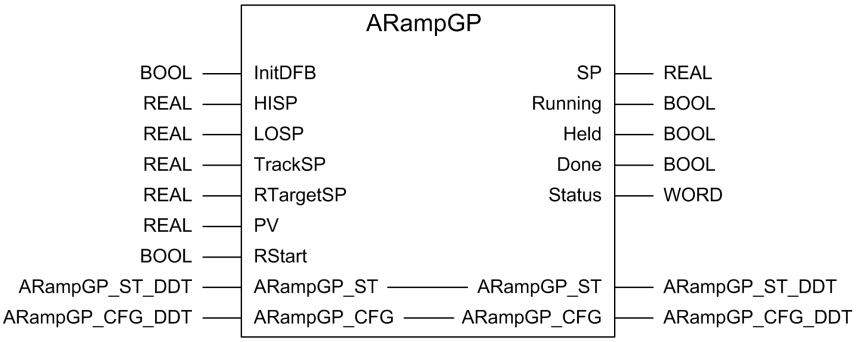
Overview

This chapter describes the ARampGP DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
InitDFB	BOOL	1 = When a raising edge is produced on this input, the signal connected to the TrackSP input pin is tracked.
HISP	REAL	High range of the setpoint (SP) in engineering units. Enables to set setpoints to match this maximum value if it exceeds this maximum value.
LOSP	REAL	Low range of the setpoint (SP) in engineering units. Enables to set setpoints to match this minimum value if it falls below it.
TrackSP	REAL	Tracking setpoint in engineering units that the DFB needs to generate while the ramp is not in operation. Refer to the SP output pin, page 568.
RTargetSP	REAL	Target setpoint in engineering units to be applied in the Program/Remote mode. Refer to the SP output pin, page 568.
PV	REAL	Current value of the measurement in engineering units. This input is only used for the purpose of holding the ramp if the measurement is excessively delayed in relation to the setpoint generated by the DFB itself (SP output). Refer to the ARampGP_CFG.MAXDEV output pin, page 568.
RStart	BOOL	1 = Run in the Program/Remote mode. 0 = Ramp Stop in the Program/Remote mode

Outputs

Output Parameter Description

Parameter	Type	Description
SP	REAL	Calculated value of the setpoint. In any event, this value is limited according to the range specified in the HISP and LOSP inputs. The setpoint calculation performed by the DFB is detailed in the following table:

Running	SP
OFF	TrackSP
ON	SP value is calculated based on the last value generated depending on the slopes configured (ARampGP_CFG.INCRate and DECRate) and the target setpoint. The ramp is Held if distance between the PV input and the SP is greater than the maximum deviation allowed (you can deactivate this monitoring by canceling the ARampGP_CFG.MAXDEV signal).

The following configuration data is considered according to the operating mode selected:

ARampGP_ST.CFGW.Owner	SC.Rem	target setpoint and Run command
OFF (Program)	OFF (Local)	SC.LTargetSP and SC.LStart
OFF	ON (Remote/Cascade - applicable while the Program is the Owner)	RTargetSP and RStart
ON (Operator)	-	ARampGP_CFG.LTargetSP and ARampGP_ST.CFGW.LStart

Parameter	Type	Description
Running	BOOL	1 = Indicates that the ramp is active. This signal responds to the xSTART command depending on the operating mode that is running. The Running signal calculation based on the DFB operating mode is detailed in the following table:

Owner	Rem	Running
OFF	OFF	SC.LStart
OFF	ON	RStart
ON	-	ARampGP_CFG.CFGW.LStart

Parameter	Type	Description
Held	BOOL	1 = Indicates that the ramp is Held. The ramp is Held if a maximum deviation greater than 0.0 (ARampGP_CFG.MAXDEV) has been configured and the distance between the setpoint (SP) and the measurement (PV) is greater than this deviation. While this signal is active, the setpoint is maintained at its last value and the ramp does not go on. The calculation of the Held signal based on the DFB operating mode is detailed in the following table:

ARampGP_CFG.MAXDEV	Running	Held
-	OFF	OFF
0.0	-	OFF
>0.0	ON	Difference between PV and SP greater than ARampGP_CFG.MAXDEV

Parameter	Type	Description
Done	BOOL	1 = Indicates that the setpoint (SP output) has reached the set target. The module stays active even if this signal is activated so that if the target setpoint is modified, the necessary ramp (raising or falling) is generated in order to reach it. The Done signal calculation based on the DFB operating mode is detailed in the following table:

Running	Done
OFF	OFF
ON	PV = RTargetSP

Parameter	Type	Description
Status	WORD	Status word generated by the standard Control RAMP DFB instantiated within the ARampGP DFB. Refer Control Help for more details.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
ARampGP_ST	ARampGP_ST_DDT	Provides the data necessary to monitor and/or control the DFB status.
ARampGP_CFG	ARampGP_CFG_DDT	Provides the data necessary to configure the DFB (Usually from the monitoring subsystem).

ARampGP_ST_DDT Type

The following table describes the ARampGP_ST_DDT type:

Name	Type	Description
STW	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
CFGW	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
SP	REAL	Current setpoint generated by the DFB. Refer to the SP output pin, page 568.

The following table describes the ARampGP_ST.STW word:

Bit	Name	Description
0	Rem	Refer to the SC.Rem public variable, page 570.
1	Held	Refer to the Held output pin, page 568.
2	Done	Refer to the Done output pin, page 568.
3	Running	Refer to the Running output pin, page 568.

The following table describes the ARampGP_ST.CFGW word:

Bit	Name	Description
0	Owner	Read/write access Enables to configure whether the setpoint parameters of the ramp are set by the Program (0) or by the Operator (1).
1	LStart	Read/write access Enables the ramp to be activated (1) or de-activated (0). This signal is loaded with the current DFB activation value if the Program is the DFB owner.

ARampGP_CFG_DDT Type

Name	Type	Description
LTargetSP	REAL	Read/write access This variable indicates what the target setpoint is and is calculated internally in the DFB as long as the Operator is not the owner. In which case, you can modify it from the monitoring subsystem.
INCRate	REAL	Read/write access This variable indicates the raising slope that has been configured. User can modify this variable from the monitoring subsystem.

Name	Type	Description
DECRate	REAL	Read/write access This variable indicates the falling slope that has been configured. User can modify this variable from the monitoring subsystem.
MAXDEV	REAL	Read/write access This variable indicates the maximum deviation allowed between measurement (PV) and setpoint (SP). If it is 0, this monitoring is canceled. You can modify it from the monitoring subsystem.

Public Variables

Public Variable Description

Variable	Type	Description
SC	ARampGP_SC_DDT	Provides the frequently needed data to monitor and control the DFB status from the sequential control.

ARampGP_SC_DDT Type

Name	Type	Description
LTargetSP	REAL	Read/write access Enables to assign the local target setpoint for the sequential control if the Program is the owner (Owner input/output = OFF) and the selected setpoint is the Local one (SC.Rem public variable = OFF). Otherwise, the current target setpoint is continuously copied to this variable.
SP	REAL	Read-only access. Refer to the SP output pin, page 568.
LStart	BOOL	Read/write access 1 = Allows the ramp to be activated if the Program is the owner (Owner input/output is 0) and the selected setpoint is the Local one (public variable SC.Rem is 0). Otherwise, the current activation command is continuously copied to this variable.
Owner	BOOL	Read-only access Refer to the ARampGP_ST.CFGW.Owner input/output pin, page 569.
Rem	BOOL	Read/write access 1 = Allows the DFB to be configured to remote setpoint RSP. 0 = Allows the DFB to be configured to local setpoint LSP. Only applicable if the Program is the owner.
Running	BOOL	Read-only access Refer to the Running output pin, page 568.
Held	BOOL	Read-only access Refer to the Held output pin, page 568.
Done	BOOL	Read-only access Refer to the Done output pin, page 568.

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the management of ramp.

Genies

Genie Properties

Refer to Genie Properties, page 62.

Genie Icons

Refer to Genie Icons, page 63.

Representation

The table describes the genies of the `gpl_ramp` library:

Graphic symbol	Genie name	Description
	Bar_SP_Horiz	Ramp representing the setpoint (SP) on a horizontal bar.
	Bar_SP_TargetSP_Horiz	Ramp representing the target setpoint (SP) and setpoint on a horizontal bar.
	Status	Ramp status

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

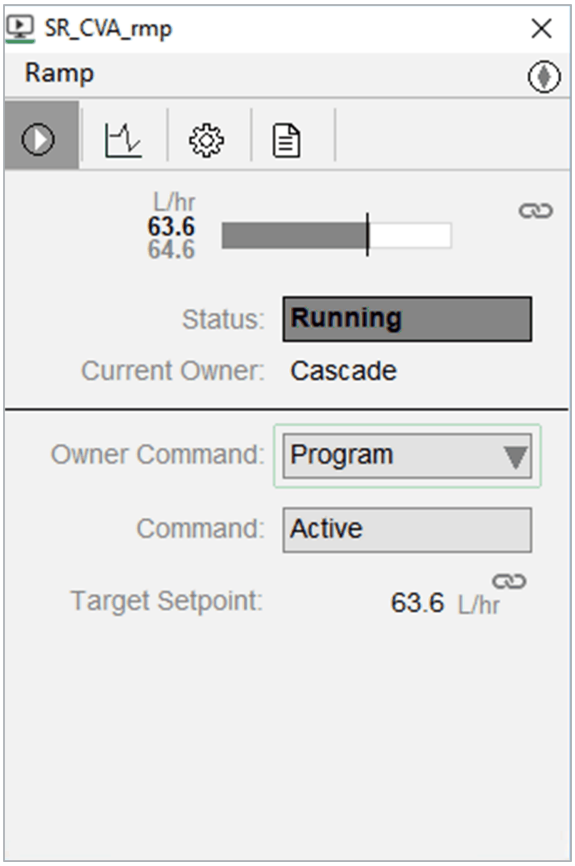
Available Tabs

During operation, clicking a ramp genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Event, page 69
  - Trend, page 69

Operator Tab

The figure shows an operator tab:



The table describes the control function of an operator tab:

Label	Item name	Description
Status	EquipStatus	Equipment status of ramp. For more details, refer to EquipStatus calculated variable, page 587.
Current Owner	Owner	Current owner of ramp, For more details, refer to operating owner, page 58.
Owner Command	OwnerSelect	Owner command of ramp (Program or Operator).
Command	Command	Command for controlling the ramp operation.
Target Setpoint	TargetSetpoint	Allows you to change the target setpoint, if owner = operator.

Engineering Tab

The figure shows an engineering tab:

The table describes the control function of engineering tab:

Label	Item name	Description	Security parameter, page 565
Raising Gradient	IncreaseRate	Raising rate of ramp value.	—
Falling Gradient	DecreaseRate	Falling rate of ramp value.	—
Max Deviation	MaximumDeviation	Maximum deviation of ramp value.	—

Items

Overview

This section describes the variables, equipstate variable, abnormal and action required conditions, alarm items, events, and trends item of `ramp`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`ProgramSetpointMode`	Setpoint source for program owner	Remote/Sequence	*Bit0*
`Held`	Held state	–	*Bit1*
`Done`	Setpoint equal target setpoint	–	*Bit2*
`Running`	Running state	–	*Bit3*
Bit items derived from ConfigurationWord.			
`OwnerSelect`	Owner of control module	Operator/Program	*Bit0*
`Command`	Command for control	–	*Bit1*
IO Items			
`StatusWord`	Status word	–	`ARampGP_ST.STW`

Item name	Description	Enumeration	Address
ConfigurationWord	Configuration word		ARampGP_ST.CFGW
TargetSetpoint	Configuration word		ARampGP_ST.CFGW
Output	Configuration word		ARampGP_ST.CFGW
MaximumDeviation	Configuration word		ARampGP_ST.CFGW
IncreaseRate	Configuration word		ARampGP_ST.CFGW
DecreaseRate	Configuration word		ARampGP_ST.CFGW
Calculated Items			
Owner	Current owner of the ramp Refer to Operating owner, page 58.	–	GPL_Owner
Equipstatus	Current status of the ramp. Refer to equipstatus calculated variable, page 587.	–	GPL_ RampEquipStatus
Equipstate	Current state of the ramp.	–	GPL_ RampEquipState
Abnormal	Shows abnormal condition of the ramp Refer to abnormal conditions, page 587.	–	GPL_ DeviceAbnormal

EquipStatus Calculated Variable

The table describes the *EquipStatus* calculated variable:

Equipment State	Conditions	Enumeration
Held	Held x Not Done x Not Running	Held
Done	Not Held x Done x Not Running	Done
Running	Not Held x Not Done x Running	Running
Inactive	Not Held x Not Done x Not Running	Inactive

Abnormal Conditions

The table describes the abnormal conditions:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
AbnormalOwner	Abnormal owner status appears, if the current owner is not a normal owner which configured at the instance level.	*Bit0*
AbnormalOperatorTab	Abnormal status of operator tab. This condition appears, if the Interlock bypass is enabled.	*Bit8*

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
`StateAlarm`	All changes in the ramp status and output.
`ActionAlarm`	All actions by the operator.

Trends

The table describes the trend items that are used by Supervision components:

Item name	Expression	Description
`Output`	`ARampGP_ST.SP`	Setpoint
`TargetSet-point`	`ARampGP_CFG.LTargetSP`	Target setpoint

Lead-Lag

What's in This Chapter

Template 576
Control 579
Supervision 583

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the lead-lag.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$LeadLagGP` template.

Description

General

The `$LeadLagGP` process control module template is used to condition the signals associated to the control of a lead-lag controller.

This process template helps you monitor and control the controller easily from the Supervision system and provides the operating modes used in the remaining EcoStruxure Process Expert process templates.

Function Description

The main functions of the `$LeadLagGP` template are described in the following table:

Function	Description
Lead-Lag	The object includes and incorporates functions provided by the Lead-Lag controller from the standard `CONT_CTL` Control Expert library.
Owner	The template manages control system level, which is the owner (Operator or Program). As a result, it is responsible for setting the setpoint to the desired position (in the Auto mode of the Lead-Lag controller) or the manual output (in the Manual mode of the Lead-Lag controller).
Interlocking	The template enables to move to the defined position if an active interlock that requires this move is detected. An interlock bypass function is available.
Setpoint	The template enables you to work under a remote (usually set from the continuous control) or local (set from the program or by the operator depending on the active owner) setpoint.

Parameters

Parameters

The `$LeadLagGP` template provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Range	Output high (EU)	`Float`	100.0	–
	Output low (EU)		0.0	–
	Output engineering unit	`String`	%	–
	Output format		####.#	–
Set-point	Force remote setpoint	`Boo-lean`	False	Enables the remote control operation for Lead-lag.
Inter-locks	For more details, refer to the topic describing the parameters of interlocks, page 81.			

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Operation	Normal owner	`Enum`	Program & Cascade (5)	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	Set owner	`Enum`	Operator (1)	Privilege to change owner. This parameter is used in Operator Tab, page 584.
	Set mode - normal		Operator (1)	Privilege to set mode normal. This parameter is used in Operator Tab, page 584.
	Set reset		Operator (Confirmed) (10)	Privilege to reset. This parameter is used in Operator Tab, page 584.
	Set parameters		Engineer (5)	Privilege to set parameters. This parameter is used in Engineering Tab, page 585.
	Set bypass conditions		Supervisor (2)	Privilege to set bypass conditions mode. This parameter is used in Interlock Tab, page 87.
	Set bypass all interlocks		Supervisor (2)	Privilege to bypass all interlocks. This parameter is used in Interlock Tab, page 87.
Asset and Display	For more details, refer to the topic describing the parameters of asset and display, page 43.			
Trend	For more details, refer to the topic describing the parameters of trend, page 47.			

Data

The table describes the data parameters:

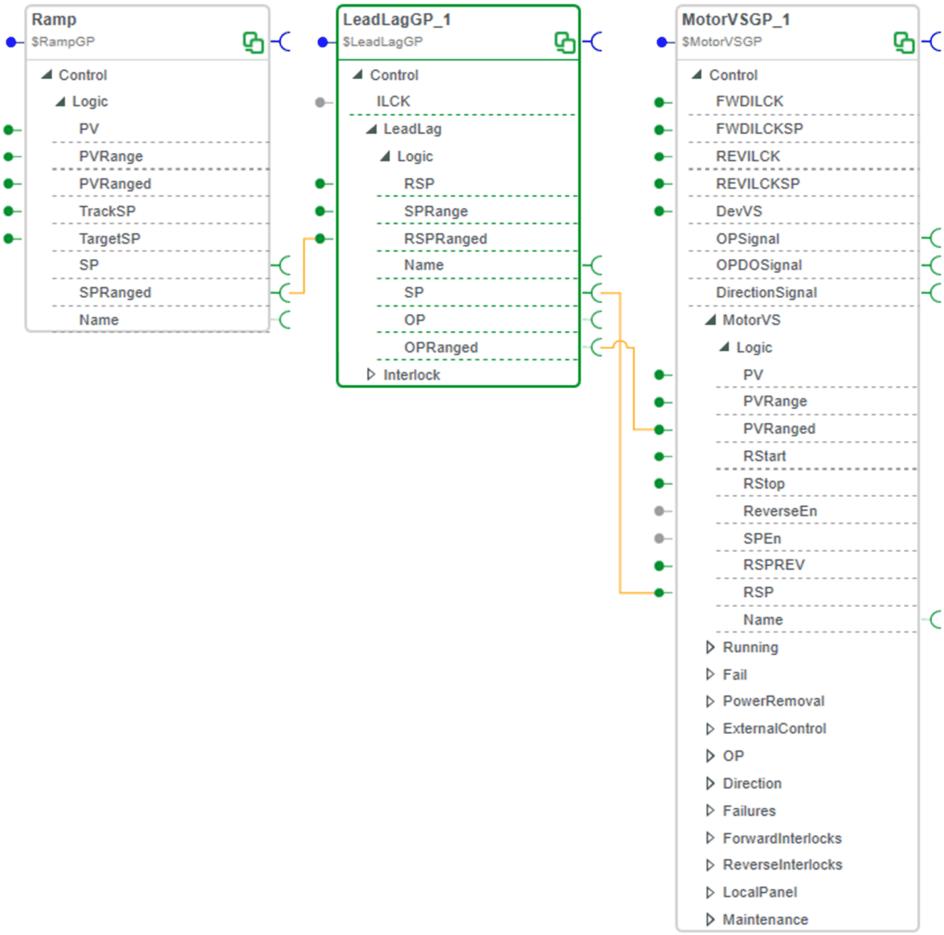
Section	Parameter description	Type	Default	Additional remarks
Configuration	Normal mode	`Enum`	Auto (0)	–
	Gain format	`String`	#####.###	–
Deadband	Deadband of Gain/lead time/ lag time/setpoint/ output	`Integer`	0	–
Events	Enable log	`Boolean`	False	Enable event logging for asset status.
Trend	Setpoint	`Boolean`	False	It disables setpoint trend tag.
	Output			It disables output trend tag.
Historize	Status/Command	`Boolean`	False	It allows to historize the change in status or commands.
	Gain/lead time/ lag time			It allows to historize the change in gain, lead time, or lag time.
	Setpoint			It allows to historize the change in setpoint.
	Output			It allows to historize the change in output.
Interlocks	For more details, refer to the topic describing the parameters of interlocks, page 81.			

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$LeadLagGP` (also applicable for `$LeadLagCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



The following link interface create the link symbol on faceplate:

- RSP or RSPRanged
- OP or Ranged

For more details, refer to Supervision chapter > Faceplate > Navigation Link of General Purpose Library User Guide.

Refer to, page 47 common control module template interface for more details.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

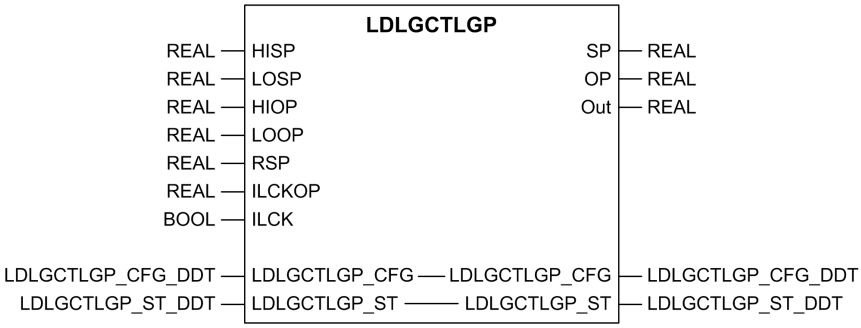
Overview

This chapter describes the `LDLGCTLGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
HISP	REAL	Upper value of the setpoint range. Used to limit the maximum value of the setpoint admitted by the controller.
LOSP	REAL	Lower value of the setpoint range. Used to limit the minimum value of the setpoint admitted by the controller.
HIOP	REAL	Upper value of the output range. Used to limit the maximum value of the output admitted by the controller.
LOOP	REAL	Lower value of the output range. Used to limit the minimum value of the output admitted by the controller.
RSP	REAL	Remote setpoint normally set by the continuous control, for example, by a PID controller output. Operates in Program/Casc mode (Automatic + Remote). Refer to the SC.REM public variable (see EcoStruxure™ Process Expert , Process Control Services User Guide).
ILCKOP	REAL	Output needs to be considered when the DFB is interlocked. Refer to the ILCK input.
ILCK	BOOL	1 = Interlocks the controller at a defined position with the ILCKOP input. Internally, the Lead-Lag controller is set to Tracking mode.

Outputs

Output Parameter Description

Parameter	Type	Description		
SP	REAL	Current setpoint that is being used by the LDLGCTLGP DFB. The setpoint calculation that the DFB performs based on the signal status is detailed in the following table:		
		Owner (OFF: Program; ON: Operator)	**Rem**	**SP** is calculated as:
		OFF	OFF	SC.LSP
		OFF	ON	RSP
		ON	-	LDLGCTLGP_ST.SP

Parameter	Type	Description		
				You can modify the setpoint directly from the monitoring subsystem while this mode is active.
OP	REAL	Output calculated in engineering units by the DFB.		
OUT	REAL	Output calculated by the DFB. Refer to the Lead-Lag controller for more details.		

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
LDLGCTLGP_CFG	LDLGCTLGP_CFG_DDT	Provides the data needed to configure the DFB normally from the monitoring subsystem or the sequential control.
LDLGCTLGP_ST	LDLGCTLGP_ST_DDT	Provides the data needed to monitor and/or control the DFB status.

LDLGCTLGP_CFG_DDT Type

Name	Type	Description
Gain	REAL	Enables to set the proportional Gain parameter of the controller.
Lead	TIME	Enables to set the derivative time parameter of the controller.
Lag	TIME	Enables to set the delay time parameter of the controller.

LDLGCTLGP_ST_DDT Type

Name	Type	Description
STW	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
CFGW	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
SP	REAL	Read/write access Refer to the SP output pin (see EcoStruxure™ Process Expert , Process Control Services User Guide). This variable indicates the current setpoint and is calculated internally in the DFB as long as the owner is not the Operator. In this case, you can modify it from the monitoring subsystem.
OP	REAL	Read/write access Refer to the OP output pin (see EcoStruxure™ Process Expert , Process Control Services User Guide). This variable indicates the current output and is calculated internally in the DFB as long as the owner is not the Operator. In this case, you can modify it from the monitoring subsystem.

LDLGCTLGP_ST.STW Word Structure

DFB status word. Read-only access. The following table describes the LDLGCTLGP_ST.STW Word structure:

Bit	Name	Description
0	Rem	Refer to the SC.Rem public variable (see EcoStruxure™ Process Expert , Process Control Services User Guide).
2	ILCK	Refer to the ILCK input pin (see EcoStruxure™ Process Expert , Process Control Services User Guide).

LDLGCTLGP_ST.CFGW Word Structure

DFB configuration word. Read/write access. The following table describes the LDLGCTLGP_ST.CFGW Word structure

Bit	Name	Description
0	Owner	Read/write access Enables to configure whether the set point is set by the Program (0) or the Operator (1).
1	ILCKBP	Read/write access Allows the interlock to be bypassed (1).
2	Mode	Read/write access Reports the operating mode of the Lead-Lag controller if the owner is the Program and enables to configure the operating mode if the owner is the Operator. 1 corresponds to Auto and 0 corresponds to Manual.

Public Variables

Public Variable Description

Variable	Type	Description
SC	LDLGCTLGP_SC_DDT	Provides the frequently needed data to monitor and control the controller status from the sequential control.

LDLGCTLGP_SC_DDT Type

Name	Type	Description
LSP	REAL	Read/write access Enables to assign the local setpoint for the sequential control if the owner is the Program (Owner is 0), the selected mode is Auto, and the selected setpoint is Local (SC.Rem is 0). Otherwise, the current setpoint (SP output) is continuously copied to this variable.
LOP	REAL	Read/write access Enables to assign the local output for the sequential control if the owner is the Program (Owner is 0) and the mode is Manual. Otherwise, the current output (OP output) is continuously copied to this variable.
Owner	BOOL	Read-only access Refer to the LDLGCTLGP_ST.CFGW.Owner input/output pin (see EcoStruxure™ Process Expert , Process Control Services User Guide).

Name	Type	Description	
Rem	BOOL	Read-only access 1 = Allows the DFB to be configured for a remote setpoint -RSP if the owner is the Program and the mode is Auto. 0 = Allows the DFB to be configured for a local setpoint -LSP if the owner is the Program and the mode is Auto.	
Man	BOOL	Write access. 1 = Allows to switch the controller to Manual mode if the owner is the Program. The DFB automatically sets the signal to 0 after processing the signal.	
Auto	BOOL	Write access 1 = Allows to switch the controller to Auto mode if the owner is the Program. The DFB automatically sets the signal to 0 after processing the signal.	
Mode	BOOL	Read-only access 1 = Auto 0 = Manual Refer to the LDLGCTLGP_ST.CFGW.Mode input/output pin (see EcoStruxure™ Process Expert , Process Control Services User Guide).	
ILCKD	BOOL	Read-only access The signal evaluation depending on the ILCK input and the LDLGCTLGP_ST.STW.ILCKBP input/output is shown in the following table:	
	ILCK	**ILCKBP**	**SC.ILCKD** is calculated as:
	OFF	-	OFF
	ON	OFF	ON
	ON	ON	OFF

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the management of lead-lag.

Genies

Genie Properties

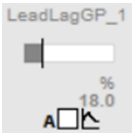
Refer to Genie Properties, page 62.

Genie Icons

Refer to Genie Icons, page 63.

Representation

The table describes the genies of the gpl_leadlag library:

Graphic symbol	Genie name	Description
	*Bar_SPOP_Horiz*	Output value status display on a horizontal bar.

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

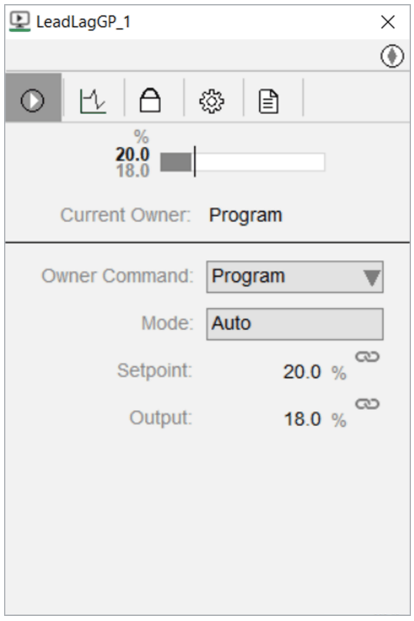
Available Tabs

During operation, clicking a lead-lag genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Event, page 69
- Tabs for optional functions, which appear only if configured:
  - Interlock, page 87

Operator Tab

The figure shows an operator tab:

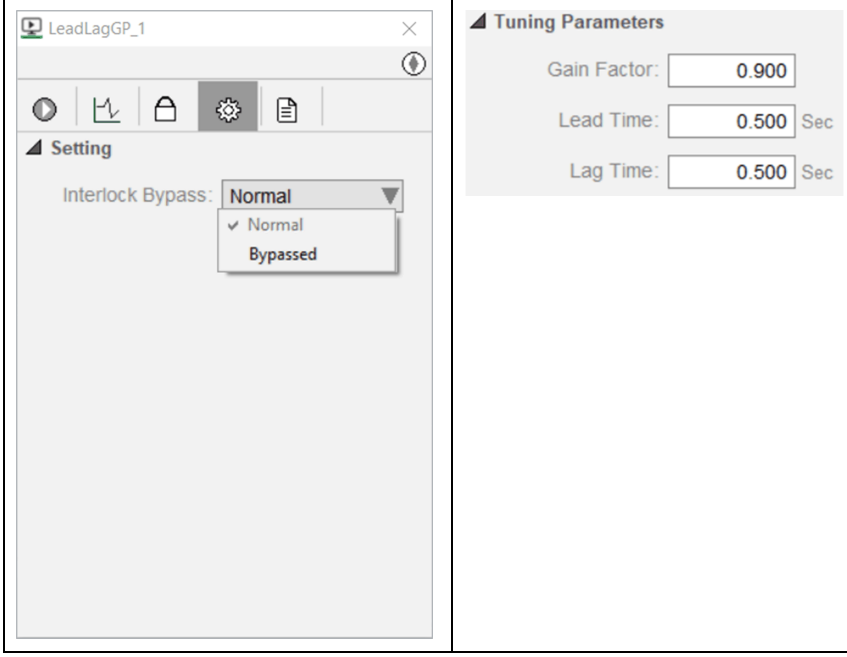


The table describes the control function of an operator tab:

Label	Item name	Description	Security parameter, page 577
Current Owner	Owner	Current owner of lead-lag, For more details, refer to operating owner, page 58.	Set owner
OwnerCom-mand	OwnerSelect	Owner command of lead-lag (Program or Operator).	Set owner
Mode	Mode	Allows you to view and change the mode. For more details, refer tooperating mode, page 59..	Set mode - normal
Set Point	CommandSet-point	Allows you to change the setpoint, if owner = operator.	Set setpoint
Output	Output	Output status of lead-lag. For more details, refer to EquipState calculated variable, page 159.	—

Engineering Tab

The figure shows an engineering tab:



The table describes the control function of engineering tab:

Label	Item name	Description	Security parameter, page 577
Interlock Bypass	InterlockBypass	Interlock bypass status Allows you to bypass interlock.	Set bypass conditions, set bypass all interlocks
Gain Factor	Gain	Gain factor	—
Lead Time	Lag	Derivative time constant	—
Lag Time	Lead	Delayed time constant	—

Items

Overview

This section describes the variables, equipstate variable, abnormal and action required conditions, alarm items, events, and trends item of `lead-lag`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`ProgramSetpointMode`	Setpoint source for program owner	Remote/Sequence	*Bit0*
`InterlockActive`	Interlock status	Interlocked/Healthy	*Bit2*
Bit items derived from ConfigurationWord.			
`OwnerSelect`	Owner of control module	Operator/Program	*Bit0*
`InterlockBypass`	Interlock bypass command	Bypassed/Normal	*Bit1*
`Mode`	lead-lag operating mode selection	Auto/Manual	*Bit2*
IO Items			
`StatusWord`	Status word	–	`LDLGCTLGP_ST.STW`
`ConfigurationWord`	Configuration word		`LDLGCTLGP_ST.CFGW`
`Gain`	Gain factor		`LDLGCTL_ LDLGCTLGP_CFG. Gain`
`Lead`	Lead time		`LDLGCTL_ LDLGCTLGP_CFG. Lead`
Lag	Lag time		`LDLGCTL_ LDLGCTLGP_CFG.Lag`
`OP`	Output		`LDLGCTL_ LDLGCTLGP_CFG.OP`
`SP`	Setpoint		`LDLGCTL_ LDLGCTLGP_CFG.SP`
Calculated Items			
`Owner`	Current owner of the lead-lag Refer to operating owner, page 58.		`GPL_Owner`
`ModeSymbol`	Current mode of the lead-lag Refer to calculated variable, page 58.		`GPLleadlagmode`
`EquipState`	Current state of the lead-lag. Refer to equipstate calculated variable, page 587.		`GPL_leadlagState`
`Abnormal`	Shows abnormal condition of the lead-lag		`GPL_Abnormal`

Item name	Description	Enumeration	Address
	Refer to abnormal conditions, page 587.		
ActionRequired	Shows action required condition of the lead-lag Refer to action required conditions, page 587.		GPL_ ActionRequired
Optional Items			
Interlocks	Refer to the topic describing the variable items of interlocks, page 60.		

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	Conditions	Enumeration
Active with Interlock	ILCK x Not ILCKBP x (Output > EnggMinScale)	Off
Passive with Interlock	(ILCK x Not ILCKBP) x Not (Output > EnggMinScale)	On
Active without Interlock	Not (ILCK x ILCKBP) x (Output > EnggMinScale)	Off
Passive without Interlock	Not (ILCK x ILCKBP) x Not Output > EnggMinScale	On Interlocked

Abnormal Conditions

The table describes the abnormal conditions:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
AbnormalOwner	Abnormal owner status appears, if the current owner is not a normal owner which configured at the instance level.	*Bit0*
AbnormalMode	Abnormal mode status appears if the current operating mode is not normal mode.	*Bit1*
AbnormalInterlockBy-pass	Abnormal interlock bypass condition appears if the Interlock bypass is enabled.	*Bit2*
AbnormalOperatorTab	Abnormal status of operator tab. This condition appears, if the Interlock bypass is enabled.	*Bit8*
AbnormalInterlockTab	Abnormal status of interlock tab.	*Bit11*
AbnormalEngineeringTab	Abnormal status of engineering tab.	*Bit12*

Action Required Conditions

The table describes the action required conditions:

Item name	Description	Address
Bit items derived from ActionRequired variable word.		
ActionInterlockTab	Action required status of interlock tab.	*Bit11*

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
`StateAlarm`	All changes in the anaolog input status and output.
`ActionAlarm`	All actions by the operator.

Trends

The table describes the trend items that are used by Supervision components:

Item name	Expression	Description
`Output`	`LDLGCTLGPGP_ST.OP`	lead-lag output value
`Setpoint`	`LDLGCTLGPGP_ST.SP`	lead-lag current setpoint

Split-Range Controller

What's in This Chapter

Template 589
Control 592
Supervision 596

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the split-range controller.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$SplitRangeGP` template.

Description

General

The `$SplitRangeGP` process control module template is used to condition the signals associated to the standard Control library SPLRG DFB.

It helps monitor and control the controller from the Supervision project and provides the operating modes used in the remaining EcoStruxure Process Expert process templates.

The standard Control library SPLRG block is used for split-range controllers (for example, cold/heat) because it allows you to generate two output signals based on a single signal, which can come, for example, from the output of a PIDGP controller.

You can link this template with the PIDCTLGP_UC and PIDMUXGP_UC process templates by using the Links Editor, when you want to control a process with two control modules (for example, two modulating valves) and PIDGP control.

Function Description

The main functions of split-range template are described in the following table:

Function	Description
SPLRG	The template includes and incorporates functions provided by the SPLRG controller from the standard `CONT_CTL` library.
Owner	The object manages control system level, which is the owner (Operator or Program). As a result, it is responsible for setting the setpoint and activation for the controller.
Interlocking	The template enables to move to the defined position if an active interlock that requires this move is detected. An interlock bypass function is available.
Setpoint	The template enables to work under a remote (normally set from the continuous control) or local (set from the program or by the operator depending on the active owner) setpoint.

Parameters

Parameters

The `$SplitRangeGP` template provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Setpoint	Force remote setpoint	`Boolean`	%	Enables the remote control operation.
Range	Output high (EU)	`Float`	100.0	–
	Output low (EU)		0.0	–
	Output engineering unit	`String`	%	–
	Output format		####.#EU	–
Interlocks	For more details, refer to the topic describing the parameters of interlocks, page 81.			

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Operation	Normal owner	`Enum`	Program & Cascade (5)	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	Set owner	`Enum`	Operator (1)	Privilege to change owner. This parameter is used in Operator Tab, page 597.
	Set setpoint		Operator (1)	Privilege to reset. This parameter is used in Operator Tab, page 597.
	Set reset		Operator (Confirmed) (10)	Privilege to change setpoint. This parameter is used in Operator Tab, page 597.
	Set Parameters		Engineer (5)	Privilege to set parameters. This parameter is used in Engineering Tab, page 598.
	Set bypass conditions		Supervisor (2)	Privilege to bypass condition. This parameter is used in Failure Tab, page 78.
	Set bypass all interlocks		Supervisor (2)	Privilege to bypass all interlocks. This parameter is used in Engineering Tab, page 598.
Asset and Display	For more details, refer to the topic describing the parameters of asset and display, page 43.			
Trend	For more details, refer to the topic describing the parameters of trend, page 47.			

Data

The table describes the data parameters:

Section	Parameter description	Type	Default	Additional remarks
Deadband	Setpoint	`Integer`	0	Deadband value for setpoint.
	Output			Deadband value for output.
Trend	Disable setpoint	`Boolean`	False	It disables setpoint trend tag.
	Disable output 1			It disables output 1 trend tag.
	Disable output 2			It disables output 2 trend tag.
Events	Enable log	`Boolean`	False	Enable event logging for asset status.
Historize	Setpoint	`Boolean`	False	It allows to historize the change in setpoint.
	Output 1			It allows to historize the change in output 1.
	Output 2			It allows to historize the change in output 2.
Interlocks	For more details, refer to the topic describing the parameters of interlocks, page 81.			

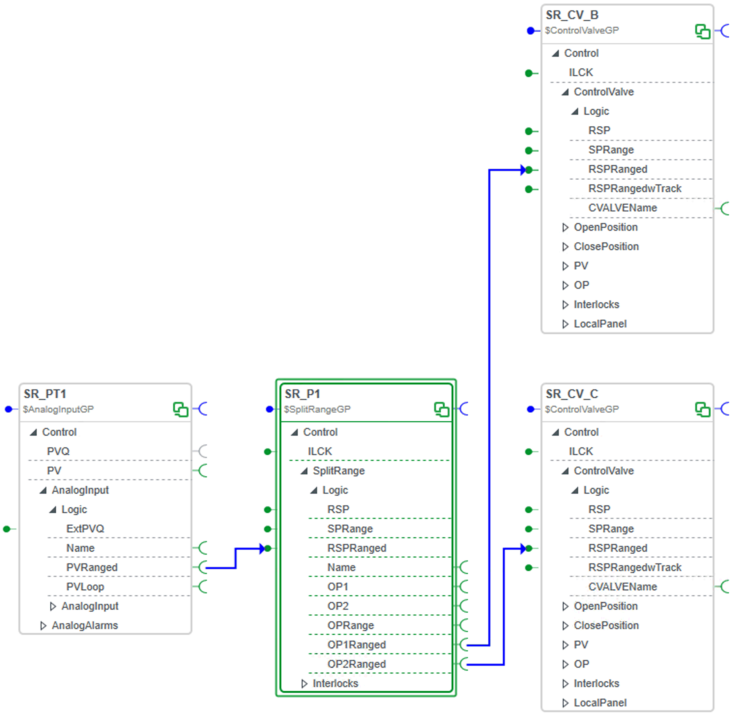
Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

Interfaces

This figure shows the `$SplitRangeGP` (also applicable for `$SplitRangeCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



The following link interface create the link symbol on faceplate:

- RSP or RSPRanged
- OP1 or OP1Ranged
- OP2 or OP2Ranged

For more details, refer to Supervision chapter > Faceplate > Navigation Link of General Purpose Library User Guide.

The table describes the control module template interfaces:

Interface identifier	Type/Role	Description	Example
`OP1`	*$Real/Def*	Output 1	–
`OP2`	*$Real/Def*	Output 2	–
`OP Range`	*$Range/Def*	output range value	–
`OP1Ranged`	*$RangedValue/Def*	Output 1 ranged value	–
`OP2Ranged`	*$RangedValue/Def*	Output 2 ranged value	–

Refer to, page 47 common control module template interface for more details on common interfaces.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

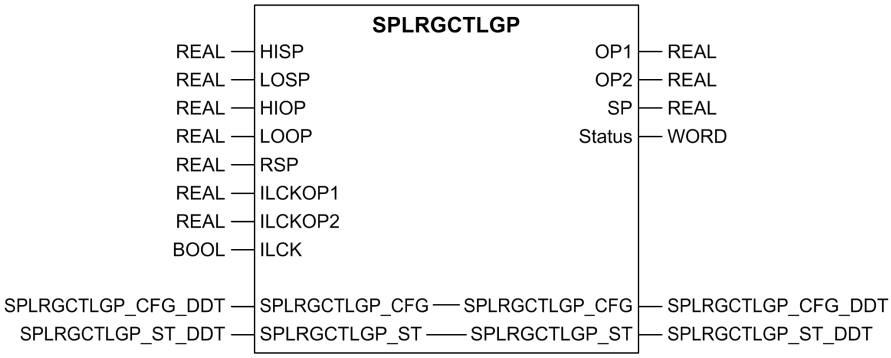
Overview

This chapter describes the `SPLRGCTLGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
HISP	REAL	High range for the setpoint (SP) in engineering units. Enables to set setpoints to this maximum value in the event that they exceed it.
LOSP	REAL	Low range for the setpoint (SP) in engineering units. Enables to set setpoints to this minimum value in the event that they fall below it.
HIOP	REAL	High range of the output (OP) in engineering units. Enables to set the output to this maximum value in the event that the output exceeds it.
LOOP	REAL	Low range of the output (OP) in engineering units. Enables to set the output to this minimum value in the event that the output falls below it.
RSP	REAL	Remote setpoint is normally set by the continuous control.
ILCKOP1	REAL	Value to which the OP1 output is forced when the DFB is interlocked. Refer to the ILCK input.
ILCKOP2	REAL	Value to which the OP2 output is forced when the DFB is interlocked. Refer to the ILCK input.
ILCK	BOOL	1 = Interlocks the device at the defined position.

Outputs

Output Parameter Description

Parameter	Type	Description
OP1	REAL	Output 1 generated by the SPLRG controller based on the parameters defined in the SPLRGCTLGP_CFG configuration structure and the setpoint range (LOSP...HISP).
OP2	REAL	Output 2 generated by the SPLRG controller based on the parameters defined in the SPLRGCTLGP_CFG configuration structure and the setpoint range (LOSP...HISP).
SP	REAL	Current setpoint. The DFB performs setpoint calculation based on the value of the inputs and of the SPLRGCTLGP_ST input/output and the setpoint calculation is detailed in the following table:

Owner (OFF: Program; ON: Operator	**Rem**	**SP** is calculated as:
OFF	OFF	SC.LSP
OFF	ON	RSP
ON	-	SPLRGCTLGP_CFG.LSP

Status	WORD	Statuses output by the SPLRG controller on the Status output. Refer to the SPLRG controller in Control for more details.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
SPLRGCTLGP_CFG	SPLRGCTLGP_CFG_DDT	Provides the data needed to configure the DFB normally from the monitoring subsystem.
SPLRGCTLGP_ST	SPLRGCTLGP_ST_DDT	Provides the data needed to monitor and/or control the DFB status.

SPLRGCTLGP_CFG_DDT Type

Name	Type	Description
OUT1_TH1	REAL	Value that the setpoint (SP) should have so that the value specified in OUT1_INF is applied on the OP1 output.
OUT1_TH2	REAL	Value that the setpoint (SP) should have so that the value specified in OUT1_SUP is applied on the OP1 output.
OUT1_INF	REAL	Value of the OP1 output when the setpoint (SP) assumes a value less than or equal to the value specified in OUT1_TH1.
OUT1_SUP	REAL	Value of the OP1 output when the setpoint (SP) assumes a value greater than or equal to the value specified in OUT1_TH2.
OUT2_TH1	REAL	Value that the setpoint (SP) should have so that the value specified in OUT2_INF is applied on the OP2 output.
OUT2_TH2	REAL	Value that the setpoint (SP) should have so that the value specified in OUT2_SUP is applied on the OP2 output
OUT2_INF	REAL	Value of the OP2 output when the setpoint (SP) assumes a value less than or equal to the value specified in OUT2_TH1.
OUT2_SUP	REAL	Value of the OP2 output when the setpoint (SP) assumes a value greater than or equal to the value specified in OUT2_TH2.

SPLRGCTLGP_ST_DDT Type

Name	Type	Description
STW	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
CFGW	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
SP	REAL	Read/write access Allows to set the local setpoint of the Operator (Owner = 1) from the monitoring subsystem. If the setpoint is set by the Program (Owner = 0), the DFB continuously assigns it the value of the current setpoint.
OP1	REAL	Read-only access Refer to the OP1 output pin, page 593.
OP2	REAL	Read-only access Refer to the OP2 output pin, page 593.

SPLRGCTLGP_ST.STW Word Structure

DFB status word. Read-only access. The following table describes the SPLRGCTLGP_ST.STW Word structure:

Bit	Name	Description
0	Rem	Refer to the SC.Rem public variable, page 595.
2	ILCK	Refer to the ILCK input pin, page 593.

SPLRGCTLGP_ST.CFGW Word Structure

DFB configuration word. Read/write access. The following table describes the SPLRGCTLGP_ST.CFGW Word structure:

Bit	Name	Description
0	Owner	Read/write access Enables to configure whether the setpoint is set by the Program (0) or by the Operator (1).
1	ILCKBP	Read/write access Allows the interlock to be bypassed (1).

Public Variables

Public Variable Description

Variable	Type	Description
SC	SPLRGCTLGP_SC_DDT	Provides the frequently needed data to monitor and control the controller status from the sequential control.

SPLRGCTLGP_SC_DDT Type

Name	Type	Description
LSP	REAL	Read/write access Enables to assign the local setpoint normally for the sequential control if the owner is the Program (Owner input/output = 0) and the selected setpoint is local (SC.Rem public variable = 1). In any other case, the DFB automatically assigns it the current value of the resulting setpoint (refer to the SP output pin, page 593.
Owner	BOOL	Read-only access Refer to the SPLRGCTLGP_ST.CFGW.Owner input/output pin, page 594.
Rem	BOOL	Read-only access 1 = Enables to configure the DFB for a remote setpoint -RSP while the owner is the Program, 0 = Enables to configure the DFB for a local setpoint -LSP while the owner is the Program.
ILCKD	BOOL	Read-only access The signal evaluation based on the ILCK input and the SPLRGCTLGP_ST.CFGW.ILCKBP input/output is shown in the following table:

Name	Type	Description		
		ILCK	ILCKBP	SC.ILCKD is calculated as:
		OFF	-	OFF
		ON	OFF	ON
		ON	ON	OFF
OP1	REAL	Read-only access Refer to the OP1 output pin, page 593.		
OP2	REAL	Read-only access Refer to the OP2 output pin, page 593.		

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the management of split-range.

Genies

Genie Properties

Refer to Genie Properties, page 62.

Genie Icons

Refer to Genie Icons, page 63.

Representation

The table describes the genies of the gpl_splitrange library:

Graphic symbol	Genie name	Description
	Indicator with_SP_OP1_OP2	The value of the setpoint and the value of the outputs (OP1 and OP2) calculated by the controller.

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

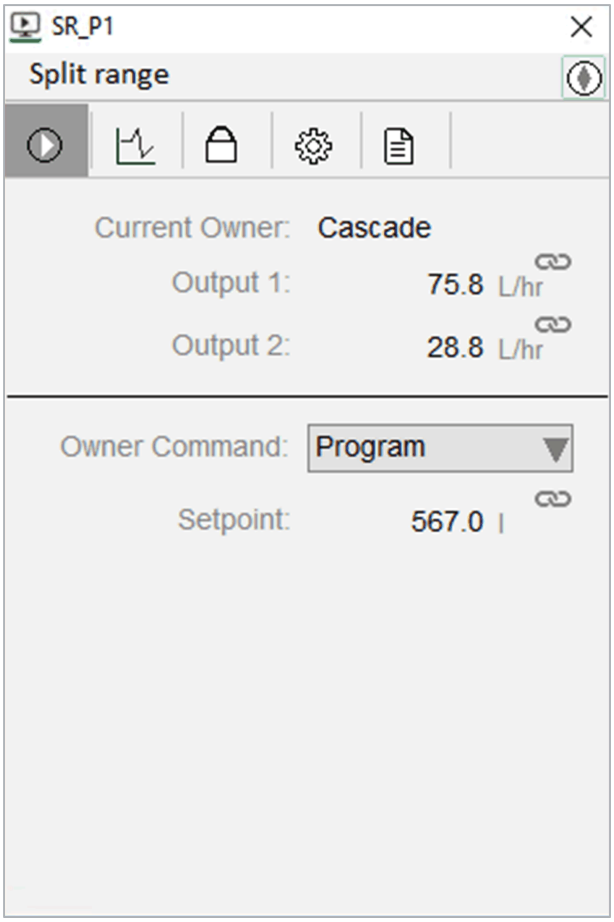
Available Tabs

During operation, clicking a split-range genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Trend, page 69
- Tabs for optional functions, which appear only if configured:
  - Interlock, page 87

Operator Tab

The figure shows an operator tab:

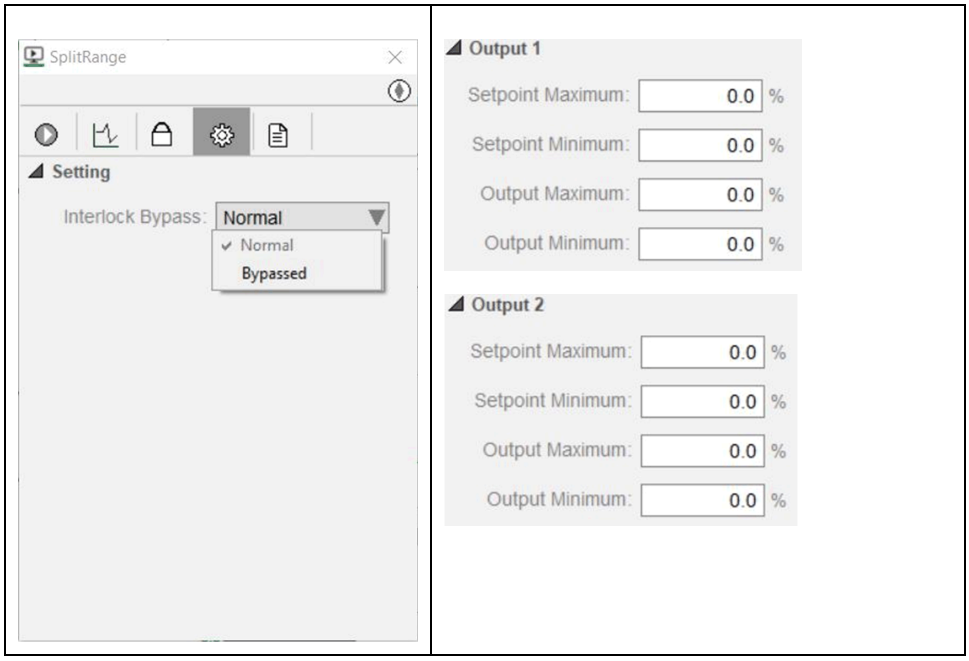


The table describes the control function of an operator tab:

Label	Item name	Description	Security parameter, page 590
Current Owner	Owner	Current owner of split-range, For more details, refer to operating owner, page 58.	—
Output 1	Output1	Output1 value.	Set parameters
Output 2	Output2	Output2 value.	Set parameters
Owner	OwnerSelect	Owner command of split-range (Program or Operator).	Set owner
Current Setpoint	CurrentSetpoint	Current setpoint of split-range.	Set setpoint

Engineering Tab

The figure shows an engineering tab:



The table describes the control function of engineering tab:

Label	Item name	Description	Security parameter, page 577
Setting			
Interlock Bypass	InterlockBypass	Interlock bypass status Allows you to bypass interlock.	Set bypass conditions, set bypass all interlocks
Output 1			
Setpoint Maximum	Output1SetpointMaximum	Setpoint for which following applies, Output1 = Output1 maximum.	Setpoint
Setpoint Minimum	Output1SetpointMinimum	Setpoint for which following applies, output1 = output1 minimum.	
Output Maximum	Output1Maximum	Upper maximum limit of the output1.	—
Output Minimum	Output1Minimum	Lower minimum limit of the output1.	—
Output 2			
Setpoint Maximum	Output2SetpointMaximum	Setpoint for which following applies, Output2 = Output2 maximum.	Setpoint
Setpoint Minimum	Output2SetpointMinimum	Setpoint for which following applies, output 2 = output2 minimum.	
Output Maximum	Output2Maximum	Upper maximum limit of the output2.	—
Output Minimum	Output2Minimum	Lower minimum limit of the output2.	—

Items

Overview

This section describes the variables, equipstate variable, abnormal and action required conditions, events, and trends item of `split-range`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
ProgramSetpointMode	Setpoint source for program owner	Remote/Sequence	*Bit0*
InterlockActive	Interlock status	Interlocked/Healthy	*Bit2*
Bit items derived from ConfigurationWord.			
OwnerSelect	Owner of control module	Operator/Program	*Bit0*
OwnerSymbol	Abnormal owner	–	*Bit1*
IO Items			
StatusWord	Status word	–	SPLRGCTLGP_ST.STW
ConfigurationWord	Configuration word		SPLRGCTLGP_ST.CFGW
Output1SetpointMaximum	Output1 Setpoint maximum limit		SPLRGCTL_SPLRGCTLGP_CFG.Out1_TH2
Output1SetpointMinimum	Output1 Setpoint minimum limit		SPLRGCTL_SPLRGCTLGP_CFG.Out1_TH1
Output1Maximum	Output1 maximum limit		SPLRGCTL_SPLRGCTLGP_CFG.Out1_SUP
Output1Minimum	Output1 minimum limit		SPLRGCTL_SPLRGCTLGP_CFG.Out1_INF
Output2SetpointMaximum	Output2 Setpoint maximum limit		SPLRGCTL_SPLRGCTLGP_CFG.Out2_TH2
Output2SetpointMinimum	Output2 Setpoint minimum limit		SPLRGCTL_SPLRGCTLGP_CFG.Out2_TH1
[Output2Maximum]	Output2 maximum		SPLRGCTL_SPLRGCTLGP_CFG.Out2_SUP
Output2Minimum	Output2 minimum		SPLRGCTL_SPLRGCTLGP_CFG.Out2_INF
Setpoint	Setpoint		SPLRGCTL_SPLRGCTLGP_ST.SP
Output1	Output1		SPLRGCTL_SPLRGCTLGP_ST.OP1
Output2	Output2		SPLRGCTL_SPLRGCTLGP_ST.OP2
Calculated Items			
Owner	Current owner of the split-range Refer to calculated variable, page 58.		GPL_Owner
Equipstate	Current state of the split-range. Refer to EquipState calculated variable, page 600.		GPL_SplitRangeEquipState

Item name	Description	Enumeration	Address
Abnormal	Shows abnormal condition of the split-range Refer to Abnormal conditions, page 600.		GPL_Abnormal
ActionRequired	Shows action required condition of the split-range Refer to Action required conditions, page 600.		GPL_SplitRangeAction-Required
Optional Items			
Interlocks	Refer to the topic describing the variable items of interlocks, page 61.		

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	Conditions	Enumeration
Interlock	ILCKActive x Not ILCKBypass	
Active	Not (ILCKActive x ILCKBypass) x (OP1>OP1Min OR OP2>Opmin)	
Passive	Not (ILCKActive x ILCKBypass) x Not (OP1>OP1Min OR OP2>Opmin)	

Abnormal Conditions

The table describes the abnormal conditions:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
AbnormalOwner	Abnormal owner status appears, if the current owner is not a normal owner which configured at the instance level.	*Bit0*
AbnormalInterlockBy-pass	Abnormal interlock bypass condition appears if the Interlock bypass is enabled.	*Bit2*
AbnormalOperatorTab	Abnormal status of operator tab. This condition appears, if the Interlock bypass is enabled.	*Bit8*
AbnormalInterlockTab	Abnormal status of interlock tab.	*Bit11*
AbnormalEngineeringTab	Abnormal status of engineering tab.	*Bit12*

Action Required Conditions

The table describes the abnormal conditions:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
ActionInterlockTab	Action required status of interlock tab.	*Bit11*

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
`StateAlarm`	All changes in the split-range status and output.
`ActionAlarm`	All actions by the operator.

Trends

The table describes the trend items that are used by Supervision components:

Item name	Description	Expression
`SP`	Setpoint	`SplitRangeGP.SP`
`OP1`	Output actuator 1	`SPLRGCTLGP_ST.OP1`
`OP1`	Output actuator 2	`SPLRGCTLGP_ST.OP1`

Step3 Controller

What's in This Chapter

Template 602
Control 605
Supervision 610

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the step3 controller.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$Step3CtlGP` template.

Description

General

The `$Step3CtlGP` process control module template is used to condition the signals associated with the control of a three-step-type controller.

It helps you to monitor and control the controller easily from the Supervision system and provides the operating modes used in the remaining EcoStruxure Process Expert process templates.

The template controls process variables through discrete control modules (for example, temperature control through hot/cold on-off valves) or is used for the positioning of analog components with discrete drives (for example, a motorized valve or a gate with position and control feedback through a two-direction motor).

Function Description

The main functions of the template are described in the following table:

Function	Description
STEP3	The template includes and incorporates functions provided by the STEP3CTLGP controller from the standard `CONT_CTL` library.
Owner	The template manages control system level, which is the owner (Operator or Program). As a result, it is responsible for setting the setpoint and activation for the controller.
Interlocking	The template enables to move to a defined setpoint if an active interlocking that requires this move is detected. An interlock bypass function is available.
Setpoint	The template enables to work under a remote (usually set from the continuous control) or local (set from the program or by the operator depending on the active owner) setpoint.
Mode	You can enable the `STEP3CTLGP` DFB (Active = 1) to operate as described in the STEP3CTLGP DFB in automatic mode, or disable (Active = 0). In this case, the DFB continues calculating but forces the outputs to 0.

Parameters

Parameters

The `$Step3CtlGP` template provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Setpoint	Force remote setpoint	`Boolean`	False	Enables the remote control operation.

Outputs (Digital Output)

The table describes the control parameters:

Name	Description
OPIncDOSignal	Digital output signal to increase PV.
OPDecDOSignal	Digital output signal to decrease PV.

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Operation	Normal owner	`Enum`	Program & Cascade (5)	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	Set owner	`Enum`	Operator (1)	Privilege to change owner. This parameter is used in Operator Tab, page 612.
	Set setpoint		Operator (1)	Privilege to change setpoint. This parameter is used in Operator Tab, page 612.
	Set reset		Operator (Confirmed) (10)	Privilege to change setpoint. This parameter is used in Operator Tab, page 612.
	Set parameters		Engineer (5)	Privilege to set parameters. This parameter is used in Engineering Tab, page 613.
	Set Bypass conditions		Supervisor (2)	Privilege to bypass condition. This parameter is used in Interlock Tab, page 87.

Section	Parameter description	Type	Default	Additional remarks
	Set bypass all interlocks		Supervisor (2)	Privilege to bypass all interlocks. This parameter is used in Engineering Tab, page 613.
Asset and Display	For more details, refer to the topic describing the parameters of asset and display, page 43.			
Trend	For more details, refer to the topic describing the parameters of trend, page 47.			

Data

The table describes the data parameters:

Section	Parameter description	Type	Default	Additional remarks
Range	Optimal range	`Enum`	Disable (0)	You can configure the optimal range: • **Disable**: Optimal range limits • **Fixed Value** - Configure optimal limits as per requirement. • **Setpoint Deviation** - Optimal range calculation based on Setpoint and Setpoint deviation.
Deadband	Present value	`Float`	0	Deadband value for present value.
Events	Enable log	`Boolean`	False	It allows to historize the change setpoints.
Trend	Disable present value	`Boolean`	False	It disables present value trend tag.
	Disable setpoint			It disables setpoint trend tag.
	Disable output			It disables output trend tag.
Historize	Present value	`Boolean`	False	It allows to historize the change in present value.
	Setpoint			It allows to historize the change in setpoint.
	Output			It allows to historize the change output.
	Optimal rang high/low			It allows to historize the optimal range high or low.
Interlocks	For more details, refer to the topic describing the parameters of interlocks, page 81.			

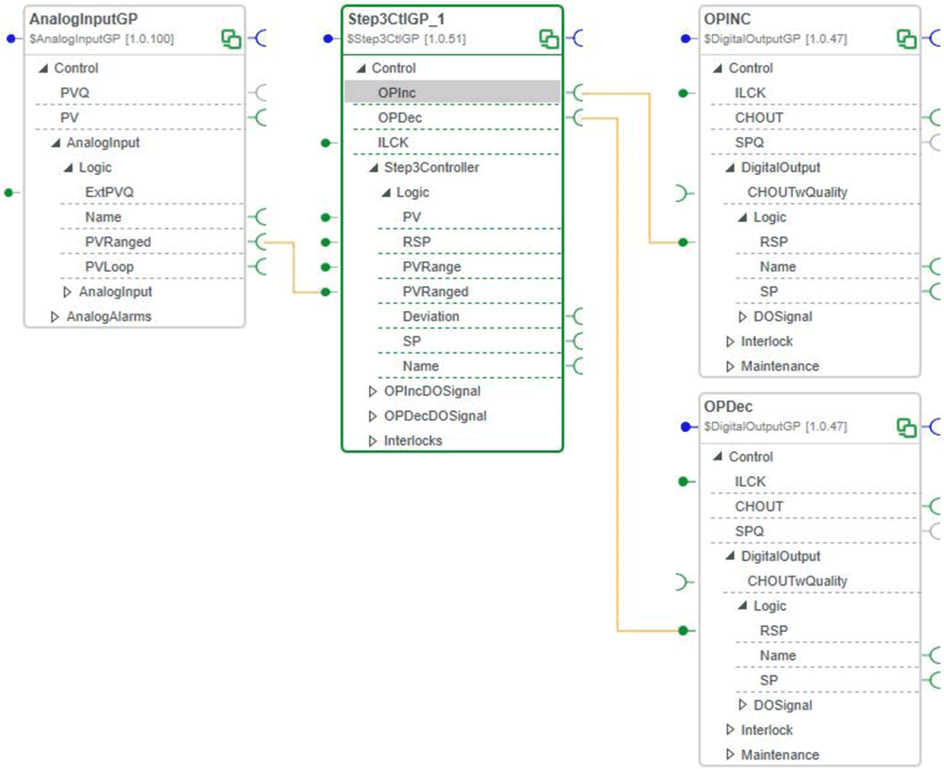
Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

Interfaces

This figure shows the `$Step3CtlGP` (also applicable for `$Step3CtlCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



The following link interface create the link symbol on faceplate:

- RSP
- PV or PVRanged
- OPInc or OPDec

The table describes the control module template interfaces:

Interface identifier	Type/Role	Description	Example
`Deviation`	*$Real/Def*	Deviation (PV-SP in step3 controller).	–
`Step3CtlName`	*$STEP3CTLIFGP/Def*	Step3 controller block instance name.	–
`OPIncVar`	*$Bool/Def*	Logical OPInc variable (instead of channel).	–
`OPDecVar`	*$Bool/Def*	Logical OPDec variable (instead of channel).	–

Refer to, page 47 common control module template interface for more details on common interfaces.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

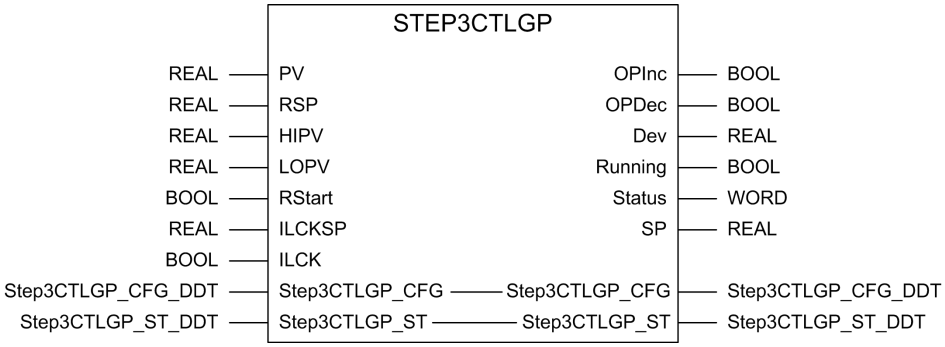
Overview

This chapter describes the `STEP3CTLGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

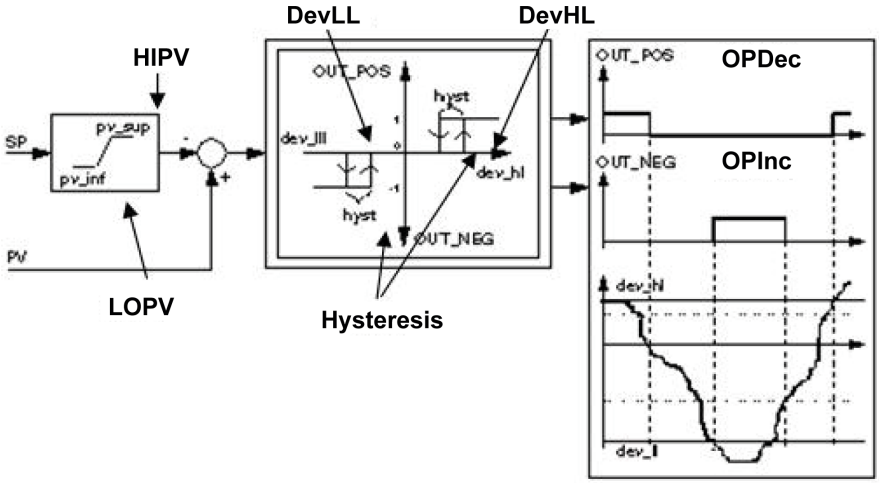
Input Parameter Description

Parameter	Type	Description
PV	REAL	Current value of the measurement in engineering units.
RSP	REAL	Remote setpoint. Is set by the continuous control. Acts in Program/ Remote mode. Refer to the SC.Rem public variable, page 609.
HIPV	REAL	Upper value of the measurement range. Used for limiting the maximum setpoint admitted by the controller. This input is internally connected to the PARA.PV_SUP parameter of the STEP3 controller.
LOPV	REAL	Lower value of the measurement range. Used for limiting the minimum setpoint admitted by the controller. This input is internally connected to the PARA.PV_INF parameter of the STEP3 controller. Check STEP3 operating graph in the following.
RStart	BOOL	1 = Enables the operation of the STEP3 DFB in Automatic mode as described in the Control documentation. With a 0 value, it forces the OPInc and OPDec outputs to 0, but the STEP3 continues to run internally and calculate the outputs. The valid Start command for the calculation is obtained from the following truth table:

Owner	Rem	Start
OFF (Program)	OFF	SC.LStart
OFF	ON	RStart
ON (Operator)	-	STEP3CTLGP_ST.CFGW.Start You can modify the setpoint directly from the monitoring subsystem while this mode is active.

ILCKSP	REAL	setpoint needs to be considered when the DFB is interlocked. Refer to the ILCK input pin.
ILCK	BOOL	1 = Forces the STEP3 DFB to run with the setpoint defined in the ILCKSP input. You can use the ILCKSP and ILCK combination to force the setpoint of a motorized valve to a pre-defined position. For example, the value corresponding to 0% aperture. In this case, the STEP3CTLGP module controls the outputs until 0% aperture is reached. If de-energizing the outputs is desired, acting on the RStart input (set to 0) is recommended. In any case, if deactivating the control signals is desirable, deactivate the control signals by interlocking the corresponding operations (using MOTORGP, MOTOR2GP, DOUTPUTGP, and so on).

STEP3 operating graph:



Outputs

Output Parameter Description

Parameter	Type	Description		
OPInc	BOOL	1 = Enabled in case of negative deviations. This output corresponds to the operation that increases PV. It is calculated according to the following table:		
		Start	**OPInc**	
		0	0	
		1	Refer to the STEP3 operation graph, page 606	
OPDec	BOOL	1 = Enabled in case of positive deviations. This output corresponds to the operation that decreases PV. It is calculated according to the following table:		
		Start	**OPDec**	
		0	0	
		1	Refer to the STEP3 operation graph, page 606	
Running	BOOL	1 = Indicates the controller is running. Read-only access.		
Dev	REAL	Deviation (PV-SP) directly extracted from the STEP3 controller.		
Status	WORD	Statuses output by the STEP3 controller on the Status output.		
SP	REAL	Current setpoint that is being considered by the STEP3 algorithm. The setpoint calculation that the DFB performs based on the signal status is detailed in the following table:		
		Owner	**Rem**	**SP**
		OFF	OFF	SC.LSP
		OFF	ON	RSP
		ON	-	STEP3CTLGP_ST.SP You can modify the setpoint directly from the monitoring subsystem while this mode is active.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
STEP3CTLGP_CFG	STEP3CTLGP_CFG_DDT	Provides the data necessary to configure the DFB usually from the monitoring subsystem or the sequential control.
STEP3CTLGP_ST	STEP3CTLGP_ST_DDT	Provides the data necessary to monitor and/or control the DFB status.

STEP3CTLGP_CFG_DDT Type

Name	Type	Description
DevLL	REAL	Enables to define the lower deviation threshold for the detected error (PV-SP). This value needs to be less than or equal to 0. Otherwise, the DFB considers it as 0. Refer to the operating graph in LOPV, page 606 and FB STEP3.
DevHL	REAL	Enables to define the higher deviation threshold for the detected error (PV-SP). This value needs to be greater than or equal to 0. Otherwise, the DFB will consider it as 0. Refer to the operating graph in LOPV, page 606 and FB STEP3.
Hysteresis	REAL	Enables to define the hysteresis value to be taken into account for calculating the OPInc and OPDec outputs by the STEP3 algorithm. This value needs to be between 0 and the minimum between DevHL and DevLL without taking the sign into account. The DFB shields erroneous hysteresis inputs with the following rules: • If CFG_Hysteresis is < 0, it is considered as 0. • If CFG_Hysteresis is > DevHL, it is considered as DevHL. If CFG_Hysteresis is > DevLL (unsigned), it is considered as DevLL (unsigned).
TargetSP	REAL	Read access This variable indicates what the current setpoint is before interlocking.

Highlight particular operation of the controller and the configuration of the three parameters above; going through the operation graph is recommended.

For example, in case of temperature regulation with a cold liquid valve and a hot liquid valve (the cold liquid valve is connected to OPDec and the hot liquid valve to OPInc), the following assumption is made:

- the setpoint is 21° (SP).
- a maximum detected error of 1° above and below with a hysteresis of 0.5% is being considered (DevHL=1, DevLL=1, Hysteresis=0.5).

In this initial case, PV-SP = -11 (<DevLL), and, as a result, OPInc=1 (output that corresponds to increasing PV).

As PV increases, the detected error decreases in absolute value until the detected error is less than -0.5, that is, DevLL+Hysteresis=-1+0.5=-0.5, when OPInc becomes 0. Verify that the output stops at the DevLL+Hysteresis threshold and not at DevLL.

Hysteresis will be achieved if the detected error drops below the DevLL value, at which point the OPInc will return to the value of 1.

On the positive side of the detected error, you can consider the event in which the temperature is 30º with the same DevHL, DevLL, and Hysteresis values.

In this initial case PV-SP = 9 (>DevHL), and, as a result, OPDec=1 (output that corresponds to decreasing PV).

As PV drops, the detected error decreases in absolute value, until the detected error is less than 0.5, that is DevHL-Hysteresis=1-0.5=0.5, at which point OPDec becomes 0. Verify that the output stops at the DevLL+Hysteresis threshold and not at DevLL.

Hysteresis will be achieved if the detected error rises above the DevLL value, at which time the OPInc will return to a value of 1.

STEP3CTLGP_ST_DDT Type

Name	Type	Description
PV	REAL	Read access Refer to the PV input pin, page 606. This variable indicates what the current measurement is.
SP	REAL	Read/write access Refer to the SP output pin, page 607. This variable indicates which is the current setpoint and is calculated internally in the DFB as long as the owner is not the Operator. In which case, you can modify it from the monitoring subsystem.
STW	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
CFGW	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.

STEP3CTLGP_ST.STW Word

Bit	Name	Description
0	Rem	Refer to the SC.Rem public variable, page 609.
2	ILCK	Refer to the ILCK input pin, page 606.
3	OPInc	Refer to the OPInc output pin, page 607.
4	OPDec	Refer to the OPDec output pin, page 607.

STEP3CTLGP_ST.CFGW Word

Bit	Name	Description
0	Owner	Read/write access Enables configuration whether the setpoint is set by the Program (0) or by the Operator (1).
1	ILCKBP	Read/write access Enables the interlock to be bypassed (1).
2	Start	Read/write access Start command while the owner is the Operator; otherwise, the valid Start is copied to ST.CFGW.Start. Refer to the RStart input pin, page 606.

Public Variables

Public Variable Description

Variable	Type	Description
SC	STEP3CTLGP_SC_DDT	Provides the frequently needed data to monitor and control the regulator status from the sequential control.

STEP3CTLGP_SC_DDT Type

The following table describes the STEP3CTLGP_SC_DDT type:

Name	Type	Description
LSP	REAL	Read/write access Enables to assign the local setpoint for the sequential control if the owner is the Program (Owner is 0), the selected mode is Auto, and the selected setpoint is Local (SC.Rem is 0). Otherwise, the current setpoint (SP output) is continuously copied to this variable.
TargetSP	REAL	Read access This variable indicates what the current setpoint is before interlocking.
PV	REAL	Read-only access Refer to the PV input pin, page 606.
LStart	BOOL	Read/write access Refer to the RStart input pin, page 606. Start command while the owner is the Program and Local; otherwise, the valid Start is copied to SC.LStart.
Owner	BOOL	Read-only access Refer to the STEP3CTLGP_ST.CFGW.Owner input/output pin, page 608.
Rem	BOOL	Read/write access 1 = Allows the DFB to be configured to remote setpoint RSP, if the owner is the Program and the mode is Auto. 0 = Allows the DFB to be configured to local setpoint LSP, if the owner is the Program and the mode is Auto.
ILCKD	BOOL	Read-only access The signal evaluation depending on the ILCK input and the STEP3CTLGP_ST.STW.ILCKBP input/output is shown in the following table:

ILCK	ILCKB-P	SC.ILCKD
OFF	-	OFF
ON	OFF	ON
ON	ON	OFF

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the management of step3 controller.

Genies

Genie Properties

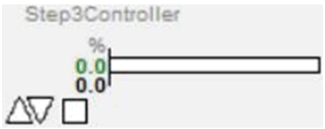
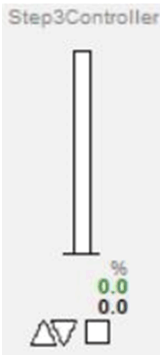
Refer to Genie Properties, page 62.

Genie Icons

Refer to Genie Icons, page 63.

Representation

The table describes the genies of the `gpl_step3control` library:

Graphic symbol	Genie name	Description
	*Bar_PVSP_Horiz*	Present value (PV) and setpoint (SP) display on a horizontal bar with engineering units.
	*Bar_PVSP_Vert*	Present value (PV) and setpoint (SP) display on a vertical bar with engineering units.
	Bar_PVSPOP_Vert_ Trend	present value, setpoint, and output display on a vertical bar with engineering units. In addition, the symbol displays trends and allows you to select the trend period in minutes.
	Display_PV	Present value display.
	Display_PVSP	Present value and setpoint display.
	Status	Status

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

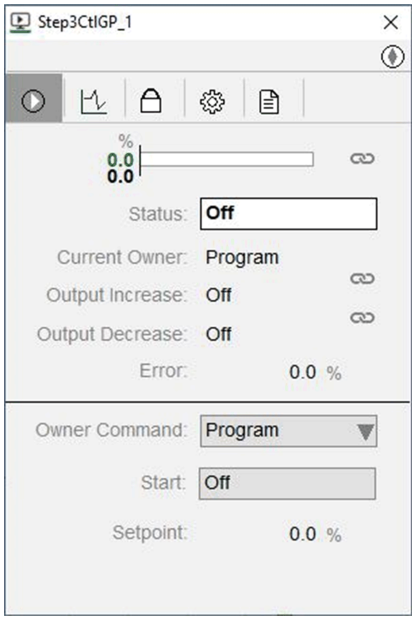
Available Tabs

During operation, clicking a step3Control genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Interlock, page 87
  - Event, page 69
  - Trend, page 69
- Tabs for optional functions, which appear only if configured:
  - Interlock, page 87

Operator Tab

The figure shows an operator tab:

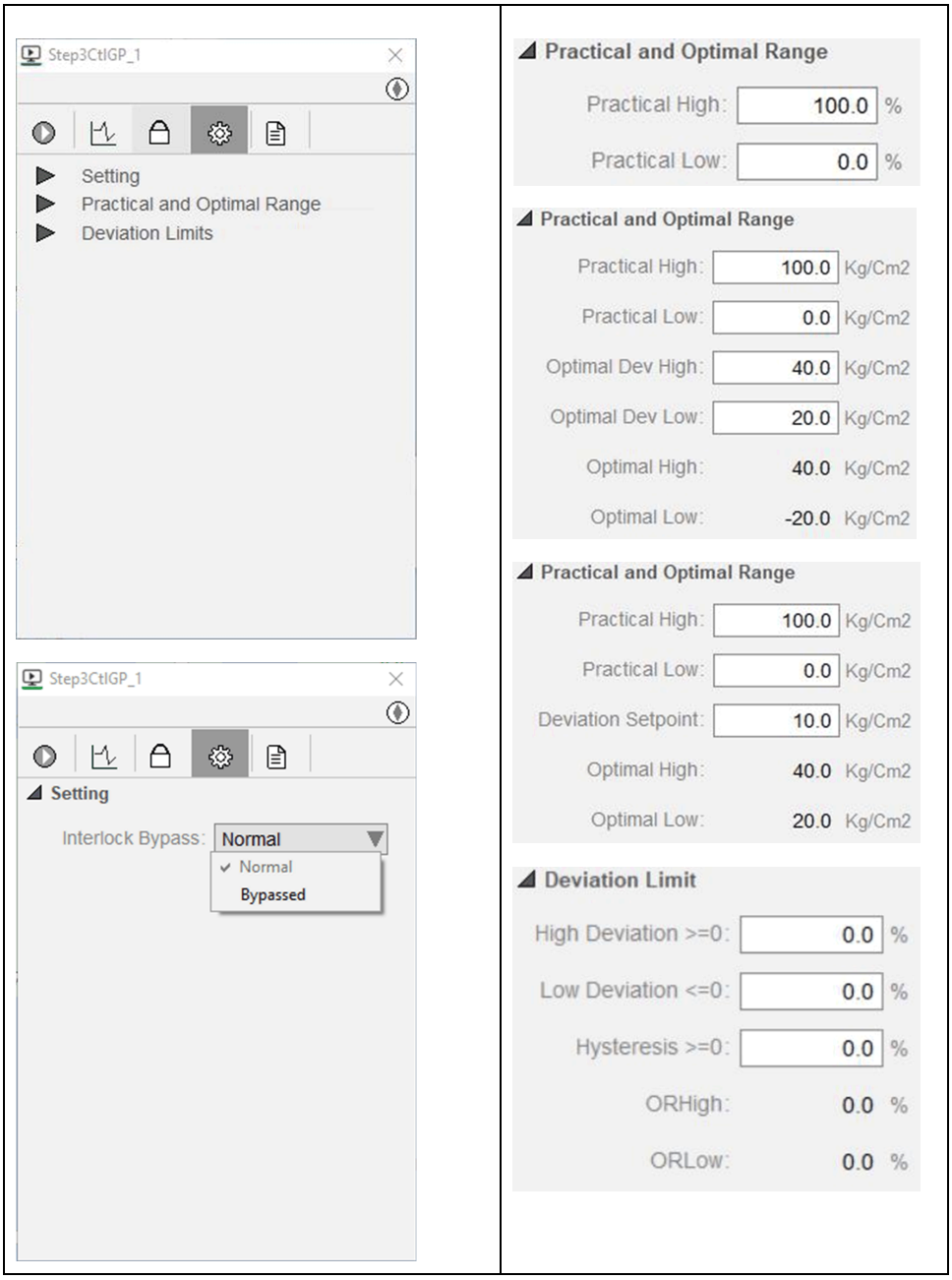


The table describes the control function of an operator tab:

Label	Item name	Description	Security parameter, page 603
Start	Start	Start state (On/Off) of step3 controller.	—
Current Owner	Owner	Current owner of step3 controller, For more details, refer to operating owner, page 58.	—
Output Increment	OutputIncrement	Logical output for negative error.	—
Output Decrement	OutputDecrement	Logical output for positive error.	—
Error	Error	Error = PV-SP	—
Owner	OwnerSelect	Owner command of step3 controller (Program or Operator).	Set owner
Start	Start	On/Off command of step3 controller in operator mode	—
Setpoint	SP	Allows you to change the setpoint, if owner = operator.	Set Setpoint

Engineering Tab

The figure shows an engineering tab:



The table describes the control function of an engineering tab:

Label	Item name	Description	Security parameter, page 603
Setting			
Interlock Bypass	InterlockBypass	Set bypass conditions, set bypass all interlocks	Set bypass conditions, set bypass all interlocks
Deviation Limits			
High deviation >=o	DeviationHigh-Setpoint	Lower deviation threshold	Set Parameter
Low deviation <=o	DeviationLowS-etPoint	Upper deviation threshold	
Hysteresis >=o	Hysteresis	Hysteresis	
Practical and Optimal Range			
Practical High	PRHigh	Practical high	Set Parameter

Label	Item name	Description	Security parameter, page 603
Practical Low	PRLow	Practical Low	
Optimal Dev High	OptimalDevia-tionHigh	Optimal deviation high	
Optimal Dev Low	OptimalDevia-tion Low	Optimal deviation high	
Deviation Setpoint	DeviationSet-point	Deviation setpoint	
Optimal High	ORHigh	Optimal range high	
Optimal Low	OrLow	Optimal range low	

Items

Overview

This section describes the variables, equipstate variable, abnormal and action required conditions, events, and trends item of `step3 controller`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`ProgramSetpointMode`	Setpoint source for program owner	Remote/Sequence	*Bit0*
`InterlockActive`	Interlock status	Interlocked/Healthy	*Bit2*
`OutputIncrement`	Output Increment	OnOff	*Bit3*
`OutputDecrement`	Output Decrement		*Bit4*
Bit items derived from ConfigurationWord.			
`OwnerSelect`	Owner of control module	Operator/Program	*Bit0*
`Start`	Start	OnOff	*Bit1*
`InterlockBypass`	Interlock bypass command	Bypassed/Normal	*Bit2*
IO Items			
`StatusWord`	Status word	–	`STEP3CTLGP_ST.STW`
`ConfigurationWord`	Configuration word		`STEP3CTLGP_ST.CFGW`
`SP`	Setpoint		`STEP3CTLGP_ST.SP`
`PV`	Present value		`STEP3CTLGP_ST.PV`
`DeviationHighSet-point`	Deviation high setpoint		`STEP3CTLGP_CFG.DevHL`
`DeviationLowSet-point`	Deviation low setpoint		`STEP3CTLGP_CFG.DevLL`
`Hysteresis`	Hysteresis		`STEP3CTLGP_CFG.Hysteresis`
`TargetSetpoint`	Target setpoint		`STEP3CTLGP_CFG.TargetSP`
Calculated Items			

Item name	Description	Enumeration	Address
`Owner`	Current owner of the step3 controller Refer to calculated variable, page 58.	–	`GPL_Owner`
`Equipstate`	Current status of the step3 controller. Refer to EquipState calculated variable, page 615.		`GPL_Step3ControlEquipState`
`Abnormal`	Shows abnormal condition of the step3 controller. Refer to Abnormal Conditions, page 615.		`GPL_Step3ControlAbnormal`
`ActionRequired`	Shows action required condition of the step3 controller. Refer to Action Required Conditions, page 616.		`GPL_ActionRequired`
Optional Items			
Interlocks	Refer to the topic describing the variable items of interlocks, page 61.		

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	Conditions	Enumeration
Interlock	`ILCKActive` x Not `ILCKBypass`	–
Active	Not (`ILCKActive` x `ILCKBypass`) x `Start`	–
Passive	Not (`ILCKActive` x `ILCKBypass`) x Not `Start`	–

Abnormal Conditions

The table describes the abnormal conditions:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
`AbnormalOwner`	Abnormal owner status appears, if the current owner is not a normal owner which configured at the instance level.	*Bit0*
`AbnormalInterlockBypass`	Abnormal interlock bypass condition appears if the Interlock bypass is enabled.	*Bit2*
`AbnormalOperatorTab`	Abnormal status of operator tab. This condition appears, if the Interlock bypass is enabled.	*Bit8*
`AbnormalInterlockTab`	Abnormal status of interlock tab.	*Bit11*
`AbnormalEngineeringTab`	Abnormal status of engineering tab.	*Bit12*

Action Required Conditions

The table describes the action required conditions:

Item name	Description	Address
Bit items derived from action required variable word.		
`ActionInterlockTab`	Action required status of interlock tab.	*Bit11*

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
`StateAlarm`	All changes in the step3 controller status and output.
`ModeAlarm`	All changes in the mode and owner.
`ActionAlarm`	All actions by the operator.

Trends

The table describes the trend items that are used by Supervision components:

Item name	Description	Expression
`PV`	Present value	`STEP3CTLGP_ST.PV`
`SP`	Setpoint	`STEP3CTLGP_ST.SP`
`OP`	Output	`STEP3CTLGP_CFG.OP`

Internal Model Controller

What's in This Chapter

Template 617
Control 621
Supervision 626

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the internal model controller.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$IMCGP` template.

Description

General

The `$IMCGP` process control module template is used to condition the signals associated with the control of a standard Control library internal model controller (IMC).

It helps monitor and control the controller easily from the Supervision system and provides the operating modes used in the remaining EcoStruxure Process Expert process templates.

The IMC internal model controller is designed to control processes with pure delays as well as nonlinear processes.

Function Description

The main functions of `$IMCGP` template are described in the following table:

Function	Description
IMC	The template includes and incorporates functions provided by the IMC controller from the standard `CONT_CTL` library. You can directly configure the IMC controller functions that have not been shown through the template interface by accessing its configuration parameters (for example, FeedForward, variable gain, and so on). Refer to the Control Help for more details.
Owner	The template manages the control system level, which is the owner (Operator or Program). As a result, it is responsible for setting the setpoint to the target position (in the Auto mode of the IMC controller) or the manual output (in the Manual mode of the IMC controller).
Interlocking	The template enables to move to the defined position if an active interlock that requires this move is detected. An interlock bypass function is available.

Function	Description
Setpoint	The template enables to work under a remote (usually set from the continuous control) or local (set from the program or by the operator depending on the active owner) setpoint.
Override	You can combine several `IMCTLGP` or other DFBs to control one single control module. This way, the DFB that is not controlling this DFB at a certain point of time works with an external output, that is, calculating the new output based on the position of control module instead of the last position calculated by the DFB itself. These functions are internally provided by the standard IMC controller.

Parameters

Parameters

The `$IMCGP` template provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Range	Output high (EU)	`Float`	100.0	–
	Output low (EU)		0.0	–
	Output engineering unit	`String`	%	–
	Output format		####.#EU	–
	Output high limit in auto mode	`Float`	100.0	–
	Output low limit in auto mode		0.0	–
Time	Sampling interval	`Duration`	00.00.01	–
	Number of delay scans after cold start	`Integer`	0	–
	Sampling period	`Duration`	00.00.01	–
	Divisor value	`Float`	1000.0	–
Setpoint	Force remote setpoint	`Boolean`	False	Enables the remote control operation for IMCTL.
Interlocks	For more details, refer to the topic describing the parameters of interlocks, page 81.			
Alarms	For parameters of LimitAlarm, page 118 and DeviationAlarm, page 124.			

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Operation	Normal owner	Enum	Program & Cascade (5)	Privilege to change setpoint This parameter is used in Operator Tab, page 630.
Security	Set owner	Enum	Operator (1)	Privilege to change owner. This parameter is used in Operator Tab, page 630.
	Set mode - normal		Operator (1)	Privilege to set mode normal. This parameter is used in Operator Tab, page 630.
	Set setpoint		Operator (1)	Privilege to change setpoint. This parameter is used in Operator Tab, page 630.
	Set mode - normal		Operator (1)	Privilege to set mode normal. This parameter is used in Operator Tab, page 630.
	Set reset		Operator (Confirmed) (10)	Privilege to reset. This parameter is used in Operator Tab, page 630.
	Set parameters		Engineer (5)	Privilege to set parameters. This parameter is used in Engineering Tab, page 631.
	Set alarm limits		Operator (1)	Privilege to set alarm limits. This parameter is used in Operator Tab, page 630.
	Set alarm acknowledge		Operator (1)	Privilege to acknowledge alarm. This parameter is used in Alarm Tab, page 67.
	Set Bypass conditions		Supervisor (2)	Privilege to set bypass conditions mode. This parameter is used in Interlock Tab, page 87.
	Set bypass all interlocks		Supervisor (2)	Privilege to bypass all interlocks. This parameter is used in Interlock Tab, page 87.
Asset and Display	For more details, refer to the topic describing the parameters of asset and display, page 43.			
Trend	For more details, refer to the topic describing the parameters of trend, page 47.			

Data

The table describes the data parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Natural time constant for open loop/closed loop process format	String	#####.###	–
	Open loop static gain		#####.###	–
	Normal mode	Enum	Auto (0)	–
Range	Optimal range	Enum	Disable (0)	It disables the optimal range indication.
	Practical range high/low disable	Boolean	True	It disables practical range high and practical low value.
	Practical range disable			It disables the practical range indication.

Section	Parameter description	Type	Default	Additional remarks
Deadband	Present value	`Integer`	0	Deadband value for present value.
	Output			Deadband value for output.
Events	Enable log	`Boolean`	False	Enable event logging for asset status.
Trend	Disable present value	`Boolean`	False	It disables present value trend tag.
	Disable setpoint			It disables setpoint trend tag.
	Disable output			It disables output trend tag.
Historize	Tuning parameters	`Boolean`	False	It allows to historize the change in tuning parameters.
	Present value			It allows to historize the change in present value.
	Setpoint			It allows to historize the change in setpoint.
	Output			It allows to historize the change in output.
Interlocks	For more details, refer to the topic describing the parameters of interlocks, page 81.			
Alarms	For parameters of LimitAlarm, page 118 and DeviationAlarm, page 124.			

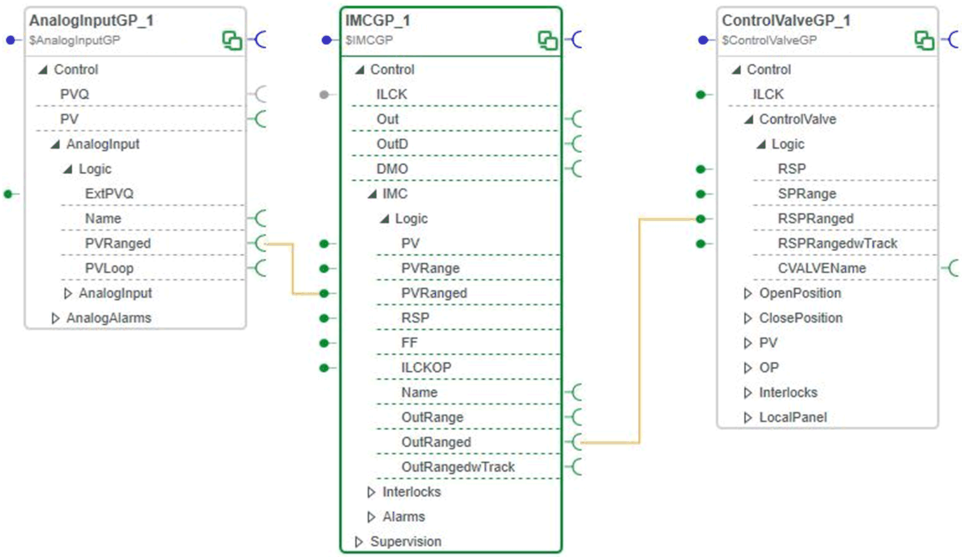
Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

Interfaces

This figure shows the `$IMCGP` (also applicable for `$IMCCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



The following link interface create the link symbol on faceplate:

- PV or PVRange or PVRanged
- Out or OutRange or OutRanged or OutRangedwTrack

The table describes the control module template interfaces:

Interface identifier	Type/Role	Description	Example
`Out`	*$Real/Def*	IMC output	–
`OutD`	*$Real/Def*	Differential output	–
`DMO`	*$Real/Def*	Module output with delay	–
`FF`	*$Real/Ref*	Feed forward value	–
`Name`	*$IMCIFGP/Def*	Instance name concatenated with control block name	–

Refer to, page 47 common control module template interface for more details on common interfaces.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

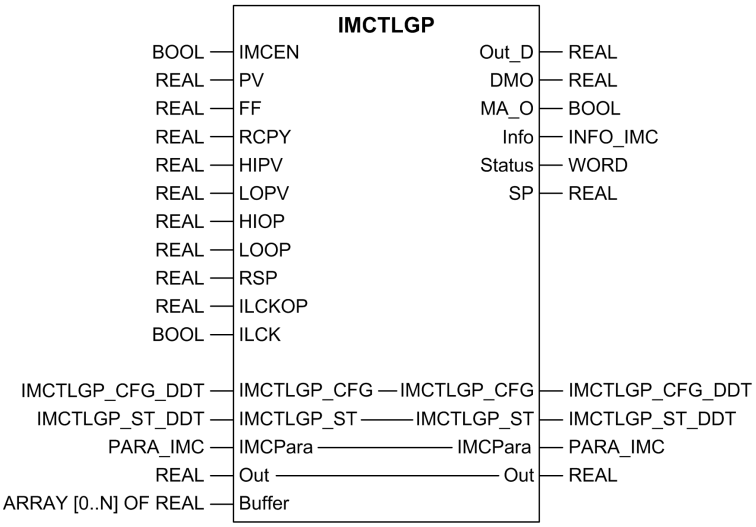
Overview

This chapter describes the `IMCTLGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
IMCEN	BOOL	1 = Enables the internal IMC DFB. The standard Control library IMC DFB needs to be periodically executed by having its EN input enabled. This input is internally connected to the IMCEN input belonging to the IMCTLGP DFB. Normally, this input is connected to a DFB that defines the execution frequency (for example, SAMPLETM). The frequency with which this input is activated should match the value that is configured on the Para.T_ECH input/output (time in seconds).
PV	REAL	Current value of the measurement in engineering units.
FF	REAL	Value of the variable that disrupts the behavior of the measurement. This input is internally connected to the FF input of the IMC controller. Refer to the Para.ff_inf, Para.ff_sup, Para.otff_inf, and Para.otff_sup parameters of the IMC controller. These values are initialized to zero by default.
RCPY	REAL	Value of the current (actual) status of the output. You can use this input to feed back the occurring output if multiple controllers controlling a single output (Override). This input is internally connected to the RCPY input of the IMC controller. Refer to the Para.EN_RCPY input.
HIPV	REAL	Upper value of the measurement range. Used for limiting the maximum setpoint admitted by the controller. This input is internally connected to the Para.PV_Sup parameter of the IMC controller (therefore, this parameter cannot be set from outside the IMCTLGP DFB because it would be overwritten).
LOPV	REAL	Lower value of the measurement range. Used for limiting the minimum setpoint admitted by the controller. This input is internally connected to the PARA.PV_INF parameter of the IMC controller (therefore, this parameter cannot be set from outside the IMCTLGP DFB because it would be overwritten).
HIOP	REAL	Upper value of the output range. Used for limiting the maximum output generated by the controller. This input is internally connected to the PARA.Out_SUP parameter of the IMC controller (therefore, this parameter cannot be set from outside the IMCTLGP DFB because it would be overwritten).
LOOP	REAL	Lower value of the output range. Used for limiting the minimum output generated by the controller. This input is internally connected to the ParaOut_INF parameter of the IMC controller (therefore, this parameter cannot be set from outside the IMCTLGP DFB because it would be overwritten).
RSP	REAL	Remote setpoint. Normally set by the continuous control, for example, by a PID controller output. Operates in Program/Cascade mode (Automatic + Remote). Refer to the SC.Rem public variable, page 625.
ILCKOP	REAL	Output needs to be used when the DFB is interlocked. Refer to the ILCK input.
ILCK	BOOL	1 = Interlocks the controller at a defined position with the ILCKOP input. Internally, the IMC controller is set to Tracking mode.
Buffer	Array [0..n] of REAL	This table of real values enables to store the measurement value for the IMC controller so that they can be delayed. The programmer needs to set the size of the table of real values that is connected to this input in such a way that the table can store the necessary data during the pure delay time of the configured process (refer to the IMCTLGP_CFG.T_Delay input/output pin, page 623). This way and depending on the configured sampling time (refer to the PARA.T_ECH input/output), it will be possible to store a sample every T_ECH seconds, a sample every 2xT_ECH seconds, and so on. Refer to the Control Help for more details.

Outputs

Output Parameter Description

Parameter	Type	Description
OutD	REAL	Differential output generated from the OutD output of the IMC controller.
DMO	REAL	Output including delay generated from the DMO output of the IMC controller.
MA_O	BOOL	1 = Current operating mode generated based on the MA_O output of the IMC controller. Refer to IMC controller for more details.
Info	Info_IMC	Variables output by the IMC controller on the Info output. Refer to the IMC controller for more details.
Status	WORD	Status output by the IMC controller on the Status output. Refer to the IMC controller for more details.
SP	REAL	Current setpoint that is being considered by the IMC algorithm. You can modify the setpoint directly from the monitoring subsystem while this mode is active.

The setpoint calculation that the DFB performs based on the signal status is detailed in the following table:

Owner (OFF: Program; ON: Operator	**Rem**	**SP is calculated as:**
OFF	OFF	SC.LSP
OFF	ON	RSP
ON	-	IMCTLGP_ST.SP You can modify the setpoint directly from the monitoring subsystem while this mode is active.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
IMCTLGP_CFG	IMCTLGP_CFG_DDT	Provides the data needed to configure the DFB normally from the monitoring subsystem or the sequential control.
IMCTLGP_ST	IMCTLGP_ST_DDT	Provides the data needed to monitor and/or control the DFB status.
IMCPara	Para_IMC	Configuration data structure of the IMC controller within the IMCTLGP DFB. Refer standard IMCTLGP DFB in the EcoStruxure Control Expert documentation for more details.
Out	REAL	Output signal of the controller. Refer standard IMCTLGP DFB in the EcoStruxure Control Expert documentation for more details.

IMCTLGP_CFG_DDT Type

Name	Type	Description
OPHiLim	REAL	Enables to define the maximum value of the output for the automatic operation of the controller. This input is internally connected to the Para.Out_MAX parameter of the IMC controller (therefore, the parameter cannot be set from outside the IMCTLGP DFB because it would be overwritten).
OPLoLim	REAL	Enables to define the minimum value of the output for the automatic operation of the controller. This input is internally connected to the ParaOut_MIN parameter of the IMC controller (therefore, the parameter cannot be set from outside the IMCTLGP DFB because it would be overwritten).
KS	REAL	Enables to set the proportional gain parameter of the controller. This input is internally connected to the Para.KP parameter of the IMC controller (therefore, the parameter cannot be set from outside the IMCTLGP DFB because it would be overwritten).
OL_Time	REAL	Enables to define the time constant for the open-loop process. This input is internally connected to the Para.OL_Time parameter of the IMC controller (therefore, this parameter cannot be set from outside the IMCTLGP DFB because it would be overwritten).
CL_PERF	REAL	Enables to set the parameter defining the relationship between the natural time constants of the open-loop/closed-loop process. This input is internally connected to the Para.C_PERF parameter of the IMC controller (therefore, this parameter cannot be set from outside the IMCTLGP DFB because it would be overwritten).
T_Delay	REAL	Enables to set the pure delay parameter in seconds. This input is internally connected to the Para.T_Delay parameter of the IMC controller (therefore, this parameter cannot be set from outside the IMCTLGP DFB because it would be overwritten).

IMCTLGP_ST_DDT Type

Name	Type	Description
STW	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
CFGW	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
PV	REAL	Read access Refer to the PV input, page 622. This variable indicates the current measurement.
SP	REAL	Read/write access Refer to the SP output, page 623. This variable indicates the current setpoint and is calculated internally in the DFB as long as the owner is not the Operator. In this case, you can modify it from the monitoring subsystem.
OP	REAL	Read/write access This variable indicates the current output and is calculated internally in the DFB as long as the owner is not the Operator. In this case, you can modify it from the monitoring subsystem.

IMCTLGP_ST.STW Word Structure

DFB status word. Read-only access. The following table describes the IMCTLGP_ST.STW Word structure.

Bit	Name	Description
0	Rem	Refer to the SC.Rem public variable, page 625.
2	ILCK	Refer to the ILCK input, page 622.
3	Override	Enables to determine if the DFB is operating with external output (1 – Override) or not (0 – Normal). Refer to the Para.EN_RCPY input.

IMCTLGP_ST.CFGW Word Structure

DFB configuration word. Read/write access. The following table describes the IMCTLGP_ST.CFGW Word structure

Bit	Name	Description
0	Owner	Read/write access Enables to configure whether the setpoint is set by the Program (0) or the Operator (1).
1	ILCKBP	Read/write access Allows the interlock to be bypassed (1).
2	Mode	Read/write access Reports the Operating mode of the IMC controller if the owner is the Program, and enables the Operating mode to be configured if the owner is the Operator. 1 corresponds to Auto and 0 to Manual.
3	Actio	Read/write access Enables you to configure the action of the IMC controller. 0 corresponds to Reverse and 1 to Forward. The DFB automatically assigns the value of this signal to the Para.rev_dir signal of the IMC controller.

Public Variables

Public Variable Description

Variable	Type	Description
SC	IMCTLGP_SC_DDT	Provides the frequently needed data to monitor and control the controller status from the sequential control.

IMCTLGP_SC_DDT Type

Name	Type	Description
LSP	REAL	Read/write access Enables the sequential control to assign the local setpoint if the owner is the Program (Owner is 0), the selected mode is Auto, and the selected setpoint is Local (SC.Rem is 0). Otherwise, the current setpoint (SP output) is continuously copied to this variable.
LOP	REAL	Read/write access. Enables the sequential control to assign the local output if the owner is the Program (Owner is 0) and the mode is Manual. Otherwise, the current output (OP output) is continuously copied to this variable.
PV	REAL	Read-only access Refer to the PV input, page 622.
Owner	BOOL	Read-only access

Name	Type	Description	
		Refer to the `IMCTLGP_ST.CFGW.Owner` input/output pin, page 623.	
`Man`	`BOOL`	Write access 1 = Allows the controller to switch to Manual mode if the owner is the Program. 0 = The DFB automatically sets the signal to 0 after processing the signal.	
`Auto`	`BOOL`	Write access 1 = Allows the controller to switch to Auto mode if the owner is the Program. 1 = The DFB automatically sets the signal to 0 after processing the signal.	
`Mode`	`BOOL`	Read-only access 1 = Auto 0 = Manual Refer to the `IMCTLGP_ST.CFGW.Mode` input/output pin, page 623.	
`Rem`	`BOOL`	Read-only access 1 = Allows the DFB to be configured for a remote setpoint -`RSP`, if the owner is the Program and the mode is Auto. 0 = Allows the DFB to be configured for a local setpoint -`LSP`, if the owner is the Program and the mode is Auto.	
`ILCKD`	`BOOL`	Read-only access The signal evaluation depending on the `ILCK` input and the `IMCTLGP_ST.STW.ILCKBP` input/output is shown in the following table:	
	`ILCK`	**`ILCKBP`**	**`SC.ILCKD`** is calculated as:
	OFF	-	OFF
	ON	OFF	ON
	ON	ON	OFF

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the internal model controller.

Genies

Genie Properties

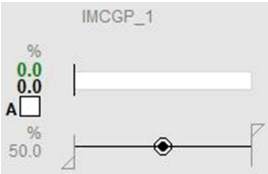
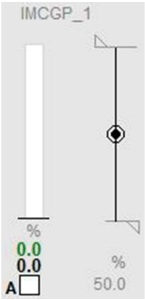
Refer to Using Genies, page 62.

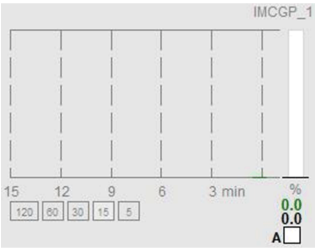
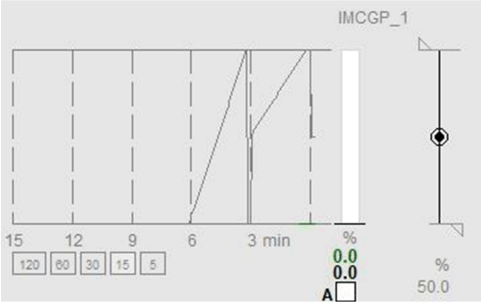
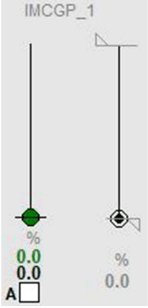
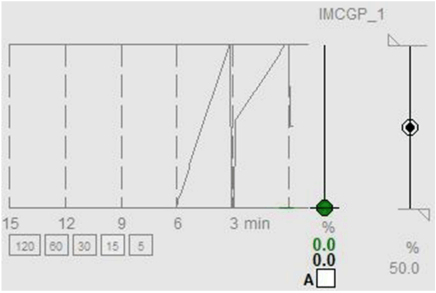
Genie Icons

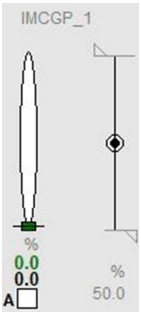
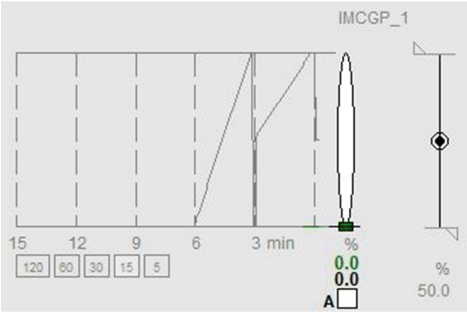
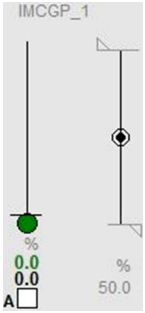
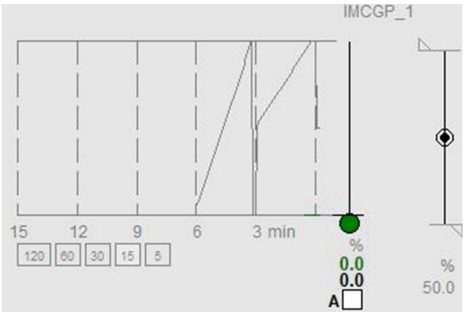
At the beginning of the manual, you can find a description of the icons, page 63 that are part of the genies.

Representation

The table describes the genies of the `gpl_IMCTL` library:

Graphic Symbol	Genie name	Description
	Bar_Op_Horiz	Horizontal bar with output (OP) display status.
	Bar_PVSP_Horiz	Horizontal bar with present value (PV) and setpoint (SP) display status.
	Bar_PVSPOP_Horiz	Horizontal bar with present value (PV), setpoint (SP), and output (OP) display status.
	Bar_OP_Vert	Vertical bar with output (OP) display status.
	Bar_PVSP_Vert	Vertical bar with present value (PV) and setpoint (SP) display status.
	Bar_PVSPOP_Vert	Vertical bar with present value (PV), setpoint (SP), and output (OP) display status.

Graphic Symbol	Genie name	Description
	Bar_PVSP_Vert_Trend	Vertical bar with present value (PV) and setpoint (SP) display status with trend.
	Bar_PVSPOP_Vert_Trend	Vertical bar with present value (PV), setpoint (SP), and output (OP) display status with trend.
	Display_PVSP	Present value (PV) and setpoint (SP) display status.
	Display_PVSPOP	Present value (PV), setpoint, and output (OP) display status.
	Flow_PVSPOP_Vert	Displays the flow on a vertical bar with engineering units, present value (PV), setpoint (SP), and output (OP).
	Flow_PVSPOP_Trend	Displays the flow on a vertical bar with engineering units, present value (PV), and setpoint (SP), Output (OP). In addition, the symbol displays trends and allows you to select the trend period in minutes.

Graphic Symbol	Genie name	Description
	Pressure_pvspop_vert	Displays the pressure on a vertical bar with engineering units, present value (PV), setpoint (SP), and output (OP).
	Pressure_PVSPOP_Trend	Displays the pressure on a vertical bar with engineering units, present value (PV), setpoint (SP), and output (OP). In addition, the symbol displays trends and allows you to select the trend period in minutes.
	Temp_PVSPOP_Vert	Displays the temperature on a vertical bar with engineering units, present value (PV), setpoint (SP), and output (OP).
	Temp_PVSPOP_Trend	Displays the temperature on a vertical bar with engineering units, present value (PV), setpoint (SP), and output (OP). In addition, the symbol displays trends and allows you to select the trend period in minutes.

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

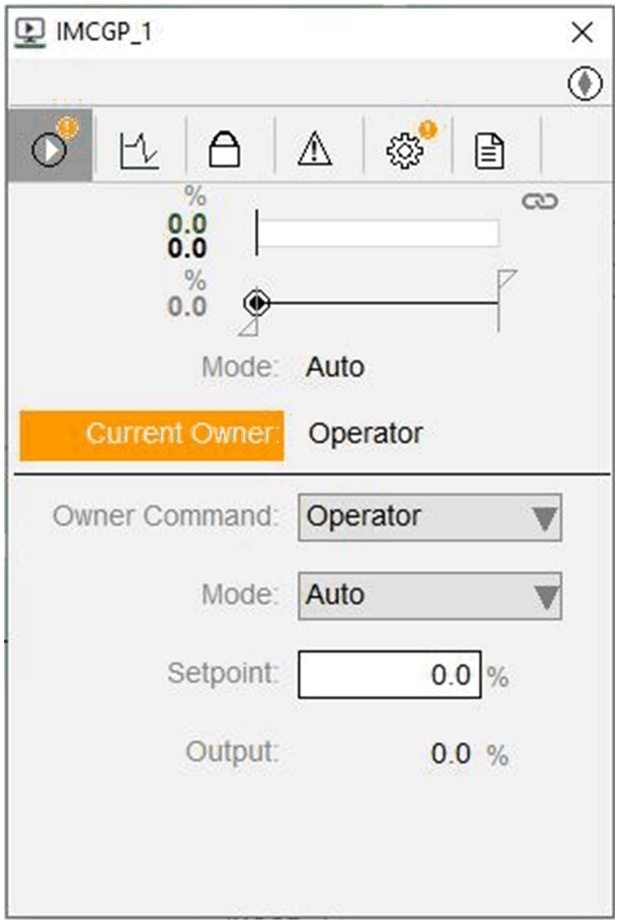
Available Tabs

During operation, clicking a IMCTL genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
  - Event, page 69
  - Trend, page 69

Operator Tab

The figure shows an operator tab:



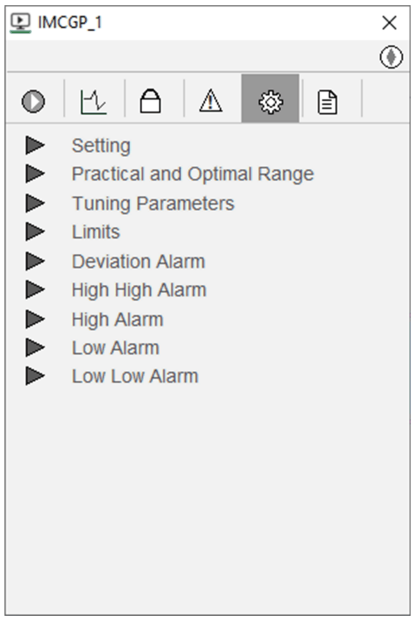
The table describes the control function of an operator tab:

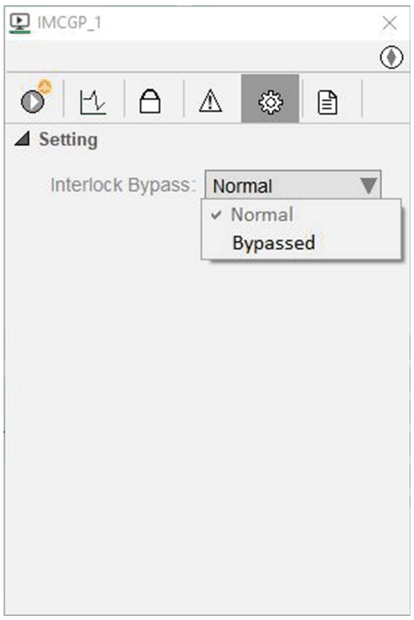
Label	Item name	Description	Security parameter, page 618
Status	EquipStatus	Equipment status of IMCTL. For more details, refer to EquipState calculated variable, page 636.	—
Mode	Mode	Allows you to view the mode (auto or manual). For more details, refer tooperating mode, page 59.	Set mode - normal
Current Owner	Owner	Current owner of IMCTL, For more details, refer to operating owner, page 58.	—
Owner Command	OwnerCommand	Owner command of IMCTL (Program or Operator).	Set owner
Mode	Mode	Allows you to view and change the mode (auto or manual). For more details, refer tooperating mode, page 59..	Set mode - normal

Label	Item name	Description	Security parameter, page 618
Setpoint	Setpoint	Allows you to change the setpoint, if owner = operator.	Set Setpoint
Output	Output	Output status of IMCTL. For more details, refer to EquipState calculated variable, page 636.	—

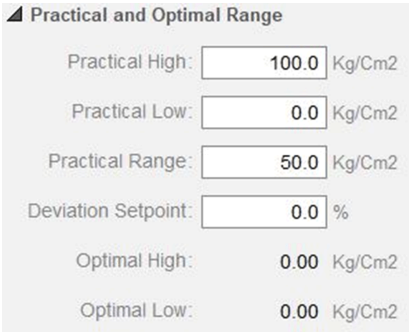
Engineering Tab

The figure shows an engineering tab:

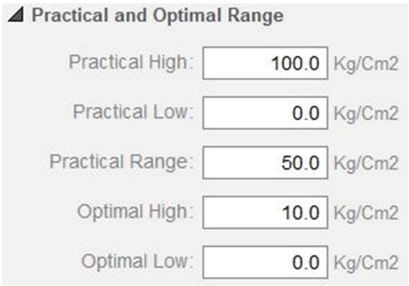


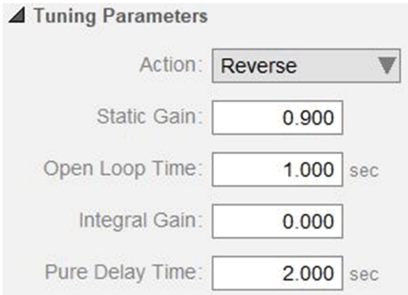


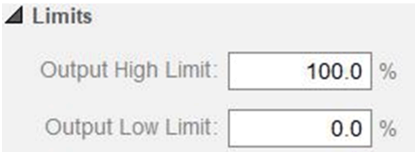
The below label appears on the faceplate when optimal range limits is set to setpoint deviation (2) and practical range disable is unselected.

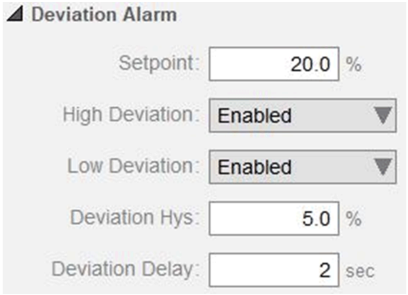


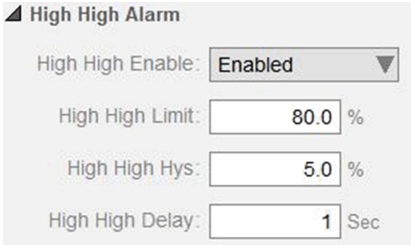
The below label appears on the faceplate when optimal range limits is set to fixed (1) value and practical range disable is unselected.

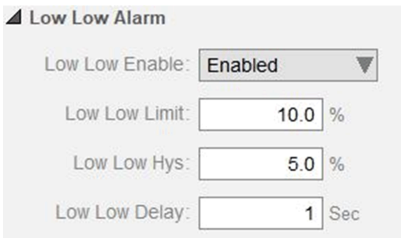


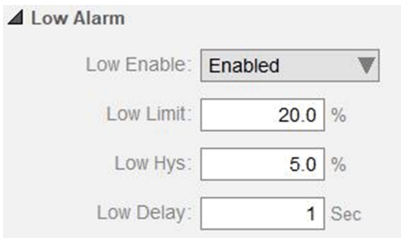












The table describes the control function of engineering tab:

Label	Item name	Description	Security parameter, page 618
Setting			
Interlock Bypass	InterlockBypass	Interlock bypass status Allows you to bypass interlock	Set bypass conditions, set bypass all interlocks
Practical and Optimal Range			
Practical High	PRHigh	Practical range high	–
Practical Low	PRLow	Practical range low	–
Practical Range	PR	Practical range	Set setpoint
Deviation Setpoint	DeviationSet-point	Maximum deviation setpoint	Set setpoint
Optimal High	ORHigh	Optimal range high	–
Optimal Low	OrLow	Optimal range low	–
Tuning Parameters			
Action	Action	Action Required (Reverse / Direct)	–
Static Gain	StaticGain	Open loop static gain	–
Open Loop Time	OpenLoopTime	Open loop time constant	–
Integral Gain	IntegralGain	Ratio between natural time constant (open loop) and required (closed loop)	–
Pure Delay Time	PureDelayTime	Pure delay of the process	–
Limits			
Output High Limit	OutputHighLimit	Output high limit	Set parameters
Output Low Limit	OutputLowLimit	Output low limit	Set parameters
Deviation Alarm			
Setpoint	Setpoint	setpoint	Set setpoint
High Deviation	HighDeviatio-nEnable	Enable high deviation.	–
Low Deviation	LowDeviatio-nEnable	Enable low deviation.	–
Deviation Hys	DeviationHyste-resis	Hysteresis value for deviation alarm.	Set parameters
Deviation Delay	DeviationDelay	Delay for enabling deviation.	Set parameters
High High Alarm			
High High Enable	HighHighEnable	High high alarm enable.	–
High High Limit	HighHighLimit	High high level setpoint.	Set setpoint

Label	Item name	Description	Security parameter, page 618
High High Hys	HighHighLimi-tHysteresis	Hysteresis high high.	Set parameters
High High Delay	HighHighLimit-Delay	Delay high high.	Set parameters
High Alarm			
High Enable	HighEnable	High alarm enable.	–
High Limit	HighLimit	High level setpoint.	Set setpoint
High Hys	HighLimitHyste-resis	Hysteresis high.	Set parameters
High Delay	HighLimitDelay	Delay high.	Set parameters
Low Low Alarm			
Low Low Enable	LowLowEnable	Low low alarm enable.	–
Low Low Limit	LowLow Limit	Low low level setpoint.	Set setpoint
Low Low Hys	LowLowLimi-tHysteresis	Hysteresis low low.	Set parameters
Low Low Delay	LowLowLimitDe-lay	Delay low low.	Set parameters
Low Alarm			
Low Enable	LowEnable	Low alarm enable.	–
Low Limit	LowLimit	Low level setpoint.	Set setpoint
Low Hys	LowLimitHyste-resis	Hysteresis low.	Set parameters
Low Delay	LowLimitDelay	Delay low.	Set parameters

Items

Overview

This section describes the variables, equipstate variable, abnormal alarm items, events, and trends item of `IMCTL`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`ProgramSetpointMode`	Setpoint source for program owner	Remote/Sequence	*Bit0*
`InterlockActive`	Interlock status	Interlocked/Healthy	*Bit2*
`OverrideOutput`	Override output selection	OnOff	*Bit3*
Bit items derived from ConfigurationWord.			
`OwnerSelect`	Owner of control module	Operator/Program	*Bit0*
`InterlockBypass`	Interlock bypass command	Bypassed/Normal	*Bit1*
`Mode`	IMCTL operating mode selection	Auto/Manual	*Bit2*

Item name	Description	Enumeration	Address
Action	Mode of IMCTL operation	Direct/Reverse	*Bit3*
IO Items			
StatusWord	Status word	–	SPLRGCTLGP_ST.STW
ConfigurationWord	Configuration word		SPLRGCTLGP_ST.CFGW
PV	Present value	–	IMCTLGP_ST.PV
Setpoint	Setpoint		IMCTLGP_ST.SP
OP	Output	–	IMCTLGP_ST.OP
OutputHighLimit	Output high limit		Output high limit
OutputLowLimit	Output low limit	–	IMCTLGP_CFG.OPLoLimit
StaticGain	Static gain		IMCTLGP_CFG.KS
OpenLoopTime	Open loop time	–	IMCTLGP_CFG.OLTime
IntegralGain	Integral gain		IMCTLGP_CFG.CL_PERF
PureDelayTime	Pure delay time	–	IMCTLGP_CFG.T_Delay
Abnormal	Abnormal		IMC_Abnormal
ActionRequired	ActionRequired	–	IMC_ActionRequired
ORDsp	-		TRUE
ORHigh	Optimal high	–	IMCTLGP_ORHigh
ORLow	Optimal low		IMCTLGP_ORLow
PR	Practical range	–	IMCTLGP_PR
PRHigh	Practical range high	–	IMCTLGP_PRHigh
PRLow	Practival range low		IMCTLGP_PRLow
OptimalDevHigh	Optimal deviation high	–	IMCTLGP_OptimalDevHigh
OptimalDevLow	IMCTLGP_OptimalDevLow		IMCTLGP_OptimalDevLow
Calculated Items			
Owner	Current owner of the IMCTL. Refer to calculated variable, page 58.		GPL_Owner
EquipState	Output information For more details, refer to EquipState calculated variable, page 636.	–	GPL_IMCTLState
ModeSymbol	Current mode of the IMCTL. Refer to calculated variable, page 59.		IMCTLGP_ModeSymbol
Abnormal	Shows abnormal condition of IMCTL. Refer to Abnormal Conditions, page 636.		GPL_Abnormal

Item name	Description	Enumeration	Address
Optional Items			
Interlocks	Refer to the topic describing the variable items of interlocks, page 61.		

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	Conditions	Enumeration
Active with Interlock	`ILCK` x `OP`	On Interlocked
Passive with Interlock	`ILCK` x Not `OP`	Off Interlocked
Active	Not `ILCK` x `OP`	On
Passive	Not `ILCK` x Not `OP`	Off

Abnormal Conditions

The table describes the abnormal conditions:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
`AbnormalOwner`	Abnormal owner status appears if the current owner is not a normal owner which configured at the instance level.	*Bit0*
`AbnormalMode`	Abnormal mode status appears if the current operating mode is not normal mode.	*Bit1*
`AbnormalInterlockBy-pass`	Abnormal interlock bypass condition appears if the Interlock bypass is enabled.	*Bit2*
`AbnormalAction`	Abnormal action	*Bit3*
`AbnormalHighDeviatio-nEnable`	If low alarm enable is active.	*Bit4*
`AbnormalLowLowEnable`	If low-low alarm enable is active.	*Bit5*
`AbnormalLowEnable`	If low alarm enable is active.	*Bit6*
`AbnormalLowDeviatio-nEnable`	If low alarm enable is active.	*Bit7*
`AbnormalOperatorTab`	Abnormal status of operator tab. This condition appears if the Interlock bypass is enabled.	*Bit8*
`AbnormalHighHighEnable`	If high-high alarm enable is active.	*Bit9*
`AbnormalHighEnable`	If high alarm enable is active.	*Bit10*
`AbnormalInterlockTab`	Abnormal status of interlock tab.	*Bit11*
`AbnormalEngineeringTab`	Abnormal status of engineering tab.	*Bit12*

Alarms

Alternate Services:

- LAlarm, page 123
- DAlarm, page 128

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
`StateAlarm`	All changes in the IMCTL status and output.
`ModeAlarm`	All changes in the mode and owner.
`ActionAlarm`	All actions by the operator.

Trends

The table describes the trend tags that are managed by Supervision components and their expression:

Item name	Description	Expression
`PV`	Present value	`IMCTLGPGP_ST.PV`
`SP`	Setpoint	`IMCTLGPGP_ST.SP`
`OP`	Output	`IMCTLGPGP_ST.OP`
`HighHighLimit`	High-high level set Point	`IMCTLGPGP_CFG.SPHH`
`LowLowLimit`	Low-low level set point	`IMCTLGPGP_CFG.SPLL`
`LowLimit`	Low level set point	`IMCTLGPGP_CFG.SPL`
`HighLimit`	High level set Point	`IMCTLGPGP_CFG.SPH`
`DeviationSetpoint`	Maximum deviation setpoint	`IMCTLGPGPL_CFG.SPD`

Pulse Width Modulation Controller

What's in This Chapter

Template 638
Control 641
Supervision 645

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the pulse width modulation controller.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$PWMOutputGP` template.

Description

General

The `$PWMOutputGP` process control module template is used to condition the signals associated with pulse-width modulation (PWM) control.

Function Description

The main functions of the template are described in the following table:

Function	Description
Owner	The template manages control system level, which is the owner (Operator or Program). As a result, it is responsible for setting the setpoint and activation for the controller.
Interlocking	The template enables to move to the defined position if an active interlock that requires this move is detected. An interlock bypass function is available.
Setpoint	The template enables to work under a remote (normally set from the continuous control) or local (set from the program or by the operator depending on the active owner) setpoint.

Parameters

Parameters

The `$PWMOutputGP` template provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Setpoint	Force remote setpoint	`Boolean`	False	Enables the remote control operation.

Outputs (Digital Output)

The table describes the control parameters:

Name	Description
OPIncDOSignal	Digital output signal to increase PV.
OPDecDOSignal	Digital output signal to decrease PV

For more details, refer to the topic describing the parameters of digital outputs, page 44.

Interlocks

For more details, refer to the topic describing the parameters of interlocks, page 81.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Operation	Normal owner	`Enum`	Program & Cascade (5)	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	Set owner	`Enum`	Operator (1)	Privilege to change owner. This parameter is used in Operator Tab, page 646.
	Set setpoint		Operator (1)	Privilege to change setpoint. This parameter is used in Operator Tab, page 646.
	Set mode - simulation		Engineer (5)	Privilege to change simulation mode. This parameter is used in Engineering Tab, page 647.
	Set mode - maintenance		Maintenance (3)	Privilege to change maintenance mode. This parameter is used in Engineering Tab, page 647.
	Set mode - out of service		Supervisor (2)	Privilege to change out of service mode. This parameter is used in Engineering Tab, page 647.
	Set reset		Operator (1)	Privilege to change setpoint. This parameter is used in Operator Tab, page 646.
	Set parameters		Engineer (5)	Privilege to set parameters. This parameter is used in Engineering Tab, page 647.
	Set bypass conditions		Supervisor (2)	Privilege to bypass condition. This parameter

Section	Parameter description	Type	Default	Additional remarks
				is used in Failure Tab, page 78.
	Set bypass all interlocks		Supervisor (2)	Privilege to bypass all interlocks. This parameter is used in Engineering Tab, page 647.
Asset and Display	For more details, refer to the topic describing the parameters of asset and display, page 43.			
Trend	For more details, refer to the topic describing the parameters of trend, page 47.			

Data

The table describes the data parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Period length format	`String`	###.###EU	–
Deadband	Setpoint	`Integer`	0	Deadband value for setpoint.
Events	Enable log	`Boolean`	False	Enable event logging for asset status.
Trend	Disable setpoint	`Boolean`	False	It disables setpoint trend tag.
	Disable output			It disables output trend tag.
Historize	Period length/ pulse time	`Boolean`	False	It allows to historize the in output signal period (Period length) and pulse time of output signal (Pulse time).
	Setpoint			It allows to historize the change in setpoint..
	Output			It allows to historize the change in output.
Interlocks	For more details, refer to the topic describing the parameters of interlocks, page 81.			

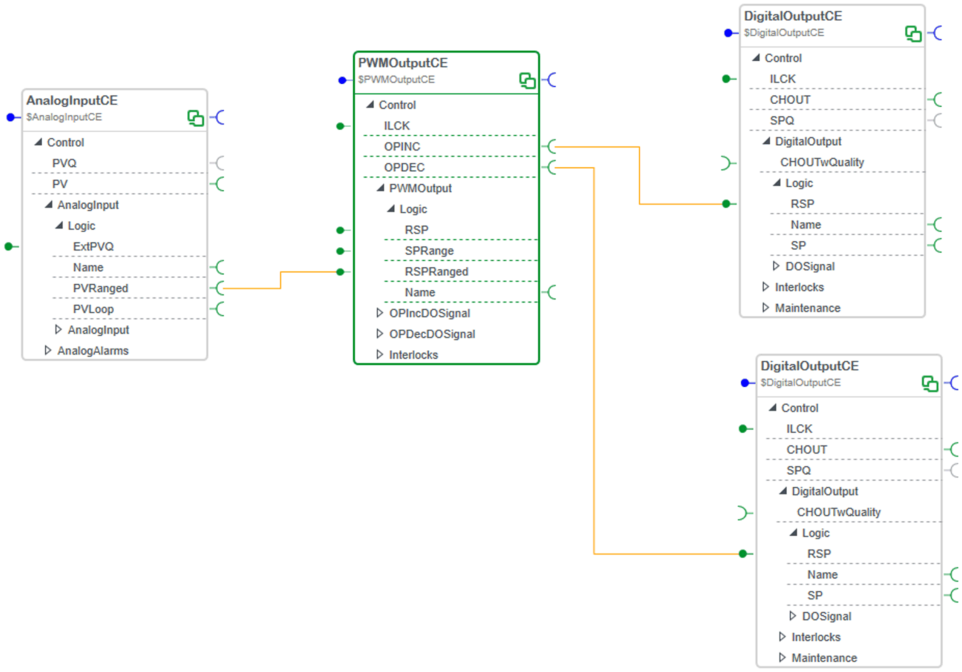
Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

Interfaces

This figure shows the `$PWMOutputCE` (also applicable for `$PWMOutputGP`) template as it appears in `Links Editor` and an example of other templates connected to it:



The following link interface create the link symbol on faceplate:

- PV or PVRange or PVRanged
- Out or OutRange or OutRanged or OutRangedwTrack

Refer to, page 47 common control module template interface for more details.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

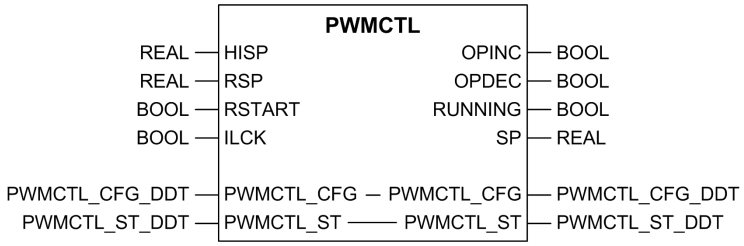
Overview

This chapter describes the `PWMCTLGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
HISP	REAL	Scale limit (positive or negative) for the setpoint range. Used to limit the maximum value of the setpoint admitted by the controller. The minimum absolute value for the setpoint is 0 just as implemented in the standard Control DFB. Therefore, if HISP > 0, the setpoint range is 0 - HISP. If HISP < 0, the setpoint range is HISP - 0. Depending on the sign of this signal and the setpoint (SP), either the output that acts to increase or decrease the measurement will be activated. Refer to the OPInc and OPDec outputs.
RSP	REAL	Remote setpoint. Usually set by the continuous control, for example, by a PID controller output. Operates in Program/Cascade mode (Automatic + Remote). Refer to the SC.REM public variable, page 644.
RStart	BOOL	1 = Enables the operation of the PWM DFB in Automatic mode as described in the Control documentation. When it has a value of 0, it forces the OPInc and OPDec outputs to 0.
ILCK	BOOL	1 = Interlocks and de-energizes the outputs OPInc and OPDec.

Outputs

Output Parameter Description

Parameter	Type	Description
OPInc	BOOL	1 = Calculates output for positive setpoints (SP > 0) by the controller.
OPDec	BOOL	1 = Calculates output for negative setpoints (SP < 0) by the controller.
LOPInc	BOOL	1 = Calculates logical output for positive setpoints (SP > 0) by the controller.
LOPDec	BOOL	1 = Calculates logical output for negative setpoints (SP < 0) by the controller.
Running	BOOL	1 = Indicates the controller is active.
SP	REAL	Current setpoint that is being considered by the controller.
Normal	BOOL	Normal mode.
OSS	BOOL	1 = Indicates the device is out of service.

The DFB performs the setpoint calculation based on the signal status. The setpoint calculation is detailed in the following table:

Owner (OFF: Program; ON: Operator)	**Rem**	**SP is calculated as:**
OFF	OFF	SC.LSP
OFF	ON	RSP
ON	–	PWMCTLGP_ST.SP You can modify the setpoint directly from the monitoring subsystem while this mode is active.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
PWMCTLGP_CFG	PWMCTLGP_CFG_DDT	Provides the data needed to configure the DFB normally from the monitoring subsystem.
PWMCTLGP_ST	PWMCTLGP_ST_DDT	Provides the data needed to monitor and/or control the DFB status.

PWMCTLGP_CFG_DDT Type

Name	Type	Description
TPeriod	TIME	Enables you to set the period for the output signal.
TMin	TIME	Enables you to set the minimum pulse time (ON or OFF).

PWMCTLGP_ST_DDT Type

Name	Type	Description
STW	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
CFGW	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
SP	REAL	Read/write access Refer to the SP output pin, page 642. This variable indicates the current setpoint and is calculated internally in the DFB as long as the owner is not the Operator. In this case, you can modify it from the monitoring subsystem.

PWMCTLGP_ST.STW Word Structure

Bit	Name	Description
0	Rem	Refer to the SC.Rem public variable, page 644.
2	ILCK	Refer to the ILCK input pin, page 642.
3	LOPInc	Refer to the LOPInc output pin, page 642.
4	LOPDec	Refer to the LOPDec output pin, page 642.

PWMCTLGP_ST.CFGW Word Structure

Bit	Name	Description
0	Owner	Read/write access Enables to configure whether the setpoint is set by the Program (0) or the Operator (1).
1	ILCKBP	Read/write access Allows the interlock to be bypassed (1).
2	Start	Read/write access

Bit	Name	Description
		Reports whether or not the controller is in operation if the owner is the Program and enables to activate/deactivate the controller if the owner is the Operator. 1 corresponds to activating and 0 to deactivating.
13	Sim	Read/write access Simulation operating mode
14	Maint	Read/write access Maintenance operating mode
15	OSS	Read/write access Out of service operating mode

Public Variables

Public Variable Description

Variable	Type	Description
SC	PWMCTLGP_SC_DDT	Provides the frequently needed data to monitor and control the controller status from the sequential control.

PWMCTLGP_SC_DDT Type

Name	Type	Description	
LSP	REAL	Read/write access Enables the sequential control to assign the local setpoint if the owner is the Program (Owner is 0), the selected mode is Auto, and the selected setpoint is Local (SC.Rem is 0). Otherwise, the current setpoint (SP output) is continuously copied to this variable.	
LStart	BOOL	Read/write access 1 = Enables the sequential control to activate/deactivate the controller if the owner is the Program (Owner is 0) and the mode is Manual. Otherwise, the current input (Start output) is continuously copied to this variable.	
Owner	BOOL	Read-only access Refer to the PWMCTLGP_ST.CFGW.Owner input/output pin, page 643.	
Rem	BOOL	Read-only access 1 = Allows the DFB to be configured for a remote setpoint -RSP, if the owner is the Program and the mode is Auto. 0 = Allows the DFB to be configured for local setpoint -LSP, if the owner is the Program and the mode is Auto.	
ILCKD	BOOL	Read-only access The signal evaluation depending on the ILCK input and the PWMCTLGP_ST.STW.ILCKBP input/output is shown in the following table:	
	ILCK	**ILCKBP**	**SC.ILCKD** is calculated as:
	OFF	-	OFF
	ON	OFF	ON
	ON	ON	OFF
Normal	BOOL	Normal mode.	

Name	Type	Description
`Sim`	`BOOL`	Read/write access Simulation operating mode, Refer to Normal output pin, page 642.
`Maint`	`BOOL`	Read/write access Maintenance operating mode
`OSS`	`BOOL`	Read/write access Out of service operating mode, Refer to OOS output pin, page 642.

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the pulse width modulated controller.

Genies

Genie Properties

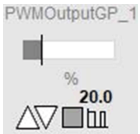
Refer to Genie Properties, page 62.

Genie Icons

Refer to Genie Icons, page 63.

Representation

The table describes the genies of the `gpl_pwmoutput` library:

Graphic symbol	Genie name	Description
	*Bar_SP_Horiz*	Setpoint (SP) on a horizontal bar.
	Status	PWM status

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

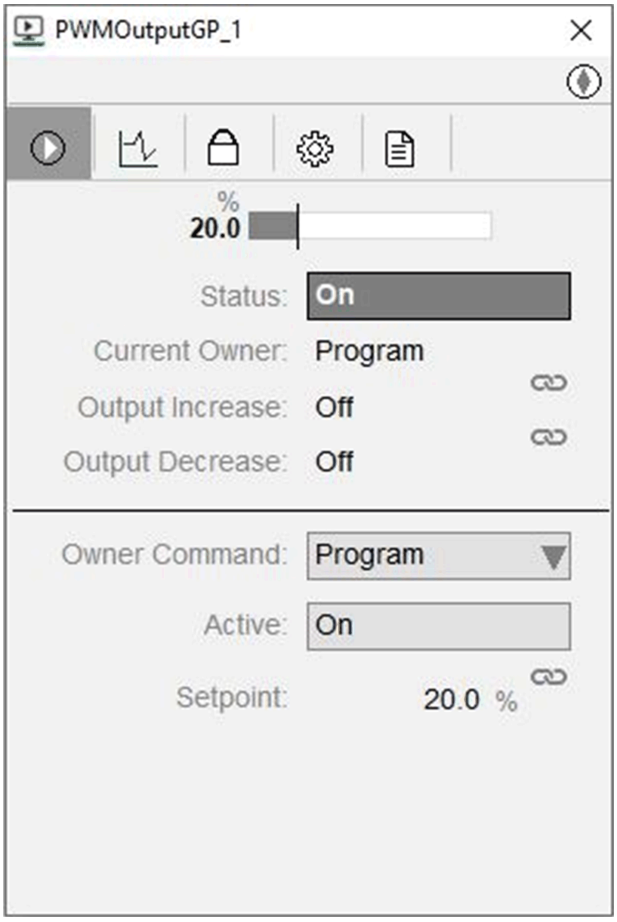
Available Tabs

During operation, clicking a PMW genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Event, page 69
  - Trend, page 69
- Tabs for optional functions, which appear only if configured:
  - Interlock, page 87

Operator Tab

The figure shows an operator tab:



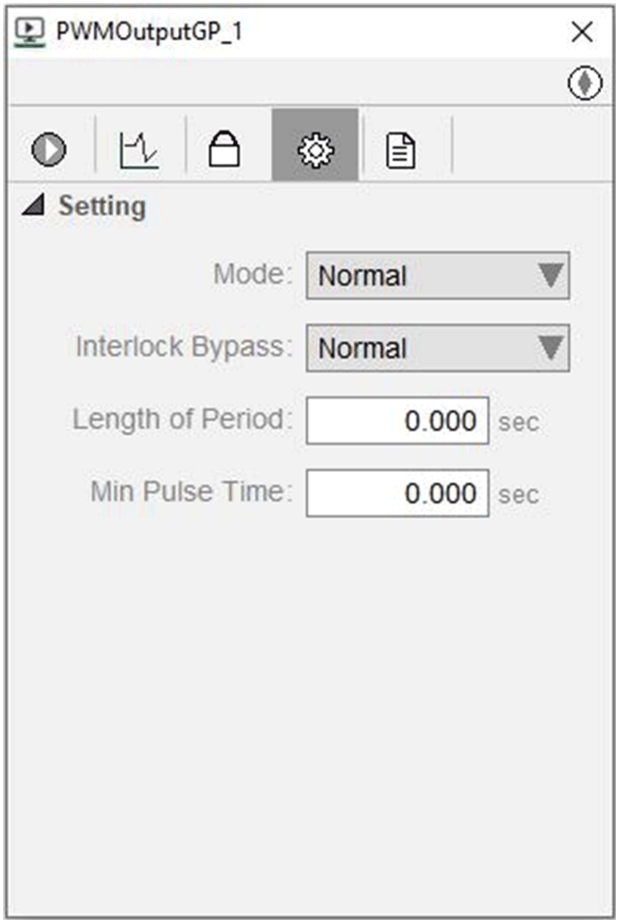
The table describes the control function of an operator tab:

Label	Item name	Description	Security parameter, page 639
Status	EquipStatus	Equipment status of PWM controller. For more details, refer to EquipStatus calculated variable, page 649.	—
Current Owner	Owner	Current owner of PWM controller, For more details, refer to operating owner, page 58.	—
Output Increase	OutputIncrement	Output increment status	—
Output Decrease	OutputDecrement	Output decrement status	—

Label	Item name	Description	Security parameter, page 639
Owner Command	Command	Owner command (Program or Operator.	Set owner
Setpoint	Setpoint	Allows you to change the setpoint, if owner = operator.	Set Setpoint
Active	Start	Control start command	—

Engineering Tab

The figure shows an engineering tab:



The table describes the control function of engineering tab:

Label	Item name	Description	Security parameter, page 639
Mode	Mode	Allows you to view and change the mode. For more details, refer to operating mode, page 59	—
Interlock Bypass	InterlockBypass	Interlock bypass status allows you to bypass interlock.	Set bypass conditions, set bypass all interlocks
Length of Period	PeriodTime	Length of output signal period	Set Parameters
Min Pulse time	MinimumPulse-Time	Minimum pulse time of output signal	Set Parameters

Items

Overview

This section describes the variables, equipstatus variable, abnormal and action required conditions, events, and trends item of PWM.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
ProgramSetpointMode	Setpoint source for program owner	Remote/Sequence	*Bit0*
InterlockActive	Interlock status	Interlocked/Healthy	*Bit2*
OutputIncrement	Output increment status	OnOff	*Bit3*
OutputDecrement	Output decrement status	OnOff	*Bit4*
Bit items derived from ConfigurationWord.			
OwnerSelect	Owner of control module	Operator/Program	*Bit0*
InterlockBypass	Interlock bypass command	Bypassed/Normal	*Bit1*
Start	Control start command	OnOff	*Bit2*
Simulation	Simulation mode	–	*Bit13*
Maintenance	Maintenance mode	–	*Bit14*
OutOfService	Device unavailable	Out Of Service/ InService	*Bit15*
IO Items			
StatusWord	Status word	–	PWMCTLGP_ST.STW
ConfigurationWord	Configuration word	–	PWMCTLGP_ST.CFGW
Setpoint	Setpoint	–	PWMCTLGP_ST.SP
PeriodTime	Length of output signal Period	–	PWMCTLGP_CFG. TPeriod
MinimumPulseTime	Minimum pulse time of output signal	–	PWMCTLGP_CFG.TMin
Calculated Items			
Owner	Current owner of the PWM. Refer to calculated variable, page 58.	–	GPL_Owner
Mode	Current mode of the PWM. Refer to mode, page 58.		GPL_ ModeSignalOutput
Equipstatus	Current status of the PWM. Refer to EquipStatus calculated variable, page 649.		GPL_ PWMOutputEquip-State

Item name	Description	Enumeration	Address
Abnormal	Shows abnormal condition of the PWM. Refer to Abnormal Conditions, page 649.		GPL_ PWMOutputAbnormal
ActionRequired	Shows action required condition of the PWM. Refer to action required conditions, page 587.		GPL_ ActionRequired
Optional Items			
Interlocks	Refer to the topic describing the variable items of interlocks, page 61.		

EquipStatus Calculated Variable

The table describes the *EquipStatus* calculated variable:

Equipment State	Conditions	Enumeration
Passive with Interlock	ILCK x Not ILCKBYPASS	Off Interlocked
Active	Not (ILCKActive x ILCKBypassx Start	On
Passive	Not (ILCKActive x ILCKBypassx NOt Start	Off

Abnormal Conditions

The table describes the abnormal conditions:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
AbnormalOwner	Abnormal owner status appears, if the current owner is not a normal owner which configured at the instance level.	*Bit0*
AbnormalMode	Abnormal mode status appears if the current operating mode is not normal mode.	*Bit1*
AbnormalInterlockBy-pass	Abnormal interlock bypass condition appears if the Interlock bypass is enabled.	*Bit2*
AbnormalOperatorTab	Abnormal status of operator tab. This condition appears, if the Interlock bypass is enabled.	*Bit8*
AbnormalInterlockTab	Abnormal status of interlock tab.	*Bit11*
AbnormalEngineeringTab	Abnormal status of engineering tab.	*Bit12*

Action Required Conditions

The table describes the abnormal conditions:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
ActionInterlockTab	Action required status of interlock tab.	*Bit11*

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
`StateAlarm`	All changes in the PWM status and output.
`ModeAlarm`	All changes in the mode and owner.
`ActionAlarm`	All actions by the operator.

Trends

The table describes the trend items that are used by Supervision components:

Item name	Expression	Description
`SP`	`PWMCTLGP_ST.SP`	Setpoint
`OP`	`PWMCTLGP_ST.OP`	Output

Sequential Control

What's in This Part

Sequential Control 652
SEQCTLGP - Sequential Control 662
SEQCTLGP - Example of Use 680
Auxiliary Functions of Sequential Control 686
SEQPARxxGP - Sequence Parameter Management 693
USERSEQxxGP - User Sequence Parameter 697

Overview

This part provides the functionality of the templates, a detailed description of the control (functions, pins, pin layout, and variables of the function blocks), and the supervision functions of sequential control.

These function blocks do not reflect any specific installation.

⚠ WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate them.
- Review the implications of communication link interruptions and take actions to mitigate them.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, and applicable codes and regulations.
- Apply local accident prevention and safety regulations and guidelines.[1]
- Test each implementation of a system for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

Sequential Control

What's in This Chapter

Description of Sequential Control.....652
Definition of Status and Commands.....653
Status Matrix.....655
Status Controller.....656
Operating Modes.....656
Using the `$SequenceDFB` Template.....657

Overview

This chapter describes the sequential control management.

Description of Sequential Control

Overview

The operation of the function block that is available for sequence control is described in this chapter.

The `SEQCTLGP` DFB belongs to the General Purpose library and is used to monitor and manage control sequences. The module is based on the ISA-S88 Standard, Part 1 (S88.01) and implements the status controller defined in this standard for controlling procedure model components (phases, operations, and so on).

Function Description

You can use this DFB to change states in the process. In compliance with ISA-S88.01, you can use this DFB in continuous processes to initiate start, stop or emergency sequences. In batch processes, this DFB can be used to set up phases and operations.

The DFB implements sequence state management and processes the commands received from:

- The Supervision system (when sequences are manually controlled by the operator)
- The batch control subsystem (in which case the subsystem sends the commands and checks the status of the sequence to determine what action to take)
- Other control sequences

For example, sequences that implement operations defined in the ISA-S88 standard can control the phases and can also be implemented through sequences managed from the component.

The table compares the functions that are available in the DFB:

Function	SEQCTLGP
Management of state machine and commands according to ISA-S88	x
Management of messages for the operator integrated with the sequence	–
Management of operating modes according to ISA-S88: Manual (enables to change from one sequence step to any other sequence step), Semi (step advance with confirmation from the operator), and Auto	x
Owner Management: Operator/Program	x

Function	SEQCTLGP
Step time automatically calculated by the control block	x
Automatic management of transitions through the SEQTRANSGP function block	x
Automatic management of sequence detected failure through the SEQFAILEDGP function block	x
Automatic management of sequence steady-step marking through the SEQSTABLESTEPGP function block	x
Possibility of displaying the sequence graphically (active step and next level of transitions and steps) from the monitoring subsystem	x
NOTE: 'x' indicates that the functions are available in the DFB.	

Program the steps and transitions that the sequences implement (through ST or SFC language) while the component (SEQCTLGP or SEQCTLGP) allows the programmer to:

- manage the statuses of the sequence and
- process the commands received from other levels as described above.

As a minimum, implement the subsequence that implements normal operation (RUNNING, page 680).

Other you can implement subsequences that attend to commands received from other levels of the control system or that react to detected failures in the process (HOLDING, RESTARTING, STOPPING, and ABORTING) detected by the sequence itself.

The table describes the main functions of the DFB:

Function	Description
Processing States and Commands	The DFB processes the commands received and determines the status of the control sequence with the objective of determining the subsequences (normal, hold, continue, abort, so on.) and which step needs to run in each cycle of the program.
Managing Entry Points in the Sequence	The DFB enables to retain the step number with which the execution of the normal sequence (Running) needs to be restarted (Restart) after being restarted from held status (Held).
Handling of Detected Failures	The component manages controller detected failures (without user intervention from the Supervision system). The component manages this detected failure and triggers the execution of the detected failure handling subsequence.

Definition of Status and Commands

Statuses

The table describes the possible sequence statuses along with the code that identifies each status:

Status	Code	Description
Idle	303	Inactive operation. Waiting for command to start (**Start**) running.
Running	302	Normal operation.
Complete	301	Normal operation has finished. Requires an initialization order (**Reset**) to return to the Idle status.
Pausing	314	A pause (**Pause**) command is received for a short period while the normal operation was being executed (Running). The normal sequence runs until it automatically goes to paused (Paused) status.
Paused	315	Status reached after the pause sequence is completed and the next steady status is reached. Upon receiving the command to continue

Status	Code	Description
		(**Resume**), the sequence continues with the next step of the normal operation sequence (Running).
Holding	316/ 326	A hold (**Hold**) command is received or a detected failure is automatically detected in the process (for example, a detected failure in a control module), and the sequence that allows the process to be changed to a known status is executed. After this sequence is finished, it automatically goes to the held (Held) status. Code 316 corresponds to having received a **Hold** and 326 to a detected failure.
Held	317/ 327	Status reached after the pause sequence is completed and the next steady status is reached. Code 317 corresponds to having received a **Hold** command and 327 to a detected failure.
Restarting	318	A restart command (**Restart**) is received while the sequence is being held (Held). The sequence runs to return to the normal operation sequence (Running). The normal operation status (Running) is set automatically after the sequence is complete.
Stopping	312	An order to stop (**Stop**) has been received. The sequence to finish the sequence in a controlled way is executed (it is not possible to continue with it). The stopped operation status (Stopped) is automatically activated after the sequence is complete.
Stopped	313	The stop sequence (Stopping) is completed. Requires an initialization command (**Reset**) to go to inactive (Idle) status.
Aborting	310	An order to end (**Abort**) has been received. A fast stop, which is not necessarily controlled is executed. The aborted operation status (Aborted) is automatically activated after the sequence is completed.
Aborted	311	An aborting sequence (Aborting) has been completed. Requires an initialization command (**Reset**) to go to inactive (Idle) status.

The (Held) status can also be reached (without receiving a command) if the control sequence itself detects a problem situation. The DFB has a signal that allows this situation to be communicated.

The DFB has the necessary means to be able to restart the Running sequence from the last steady step in which the initial run was held by a **Hold** command.

Commands

The table describes the commands that can be sent to the sequence along with the code and description of each command:

Status	Code	Description
Start	1	Allows the normal operation of the sequence (Running) to be started. It is only valid if the status is inactive (Idle).
Stop	2	Allows the sequence execution to be stopped (Running, Pausing, Paused, Holding, Held, or Restarting) and the stopping sequence to be run (Stopping).
Hold	3	Allows the sequence to be stopped (Running, Pausing, Paused, Holding, Held, or Restarting) and the stopping sequence to be run (Stopping).
Restart	5	Allows the sequence for continuing the operation (Restarting) and resume normal operation (Running).
Abort	4	Allows the sequence to be aborted (in any status except Idle, Complete, Aborting, and Aborted) and the aborting sequence to be run (Aborting).
Reset	8	Triggers the transition to idle status (in Complete, Aborted, or Stopped status).
Pause	6	Allows normal execution (Running) to be paused in the next steady Sequence status.
Resume	7	Allows normal operation of the sequence (Running) to be resumed from the paused (Paused) status.

Status Matrix

Description

The following table describes the finite status controller that the DFB implements:

Initial State	Command	Activity	Final State (without command)
Idle	Start	Running	–
Running	Stop	Stopping	Stopped
	Hold	Holding	Held
	Pause	Pausing	Paused
	Abort	Aborting	Aborted
Complete	Reset	Idle	–
Pausing	Stop	Stopping	Stopped
	Hold	Holding	Held
	Abort	Aborting	Aborted
Paused	Stop	Stopping	Stopped
	Resume		Running
	Abort	Aborting	Aborted
Holding	Stop	Stopping	Stopped
	Abort	Aborting	Aborted
Held	Stop	Stopping	Stopped
	Restart	Restarting	Running
	Abort	Aborting	Aborted
Restarting	Stop	Stopping	Stopped
	Hold	Holding	Held
	Abort	Aborting	Aborted
Stopping	Abort	Aborting	Aborted
Stopped	Reset	Idle	–
	Abort	Aborting	Aborted
Aborting	No action possible		
Aborted	Reset	Idle	–

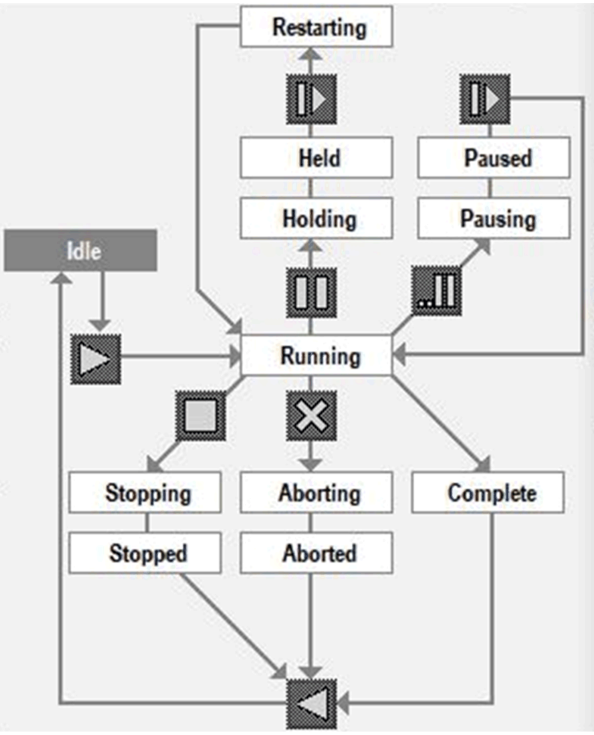
The Held status can also be reached (without receiving a command) if the control sequence itself encounters a detected error. The block has a signal that allows this situation to be communicated.

The block has the necessary means to be able to restart the Running sequence from the last steady step in which the initial run was held by a **Hold** command.

Status Controller

Description

The following diagram describes the possible statuses and commands that the Control Expert component manages according to the ISA-S88 standard:



Next to the transition arrows, the commands that the component receives (from the Supervision system or another higher level sequence) are shown in black, and the flags that the sequence itself generates are shown in red in parentheses.

Operating Modes

Mode Table

The following table provides the description of the sequence operating modes of `SEQCTLGP` DFB:

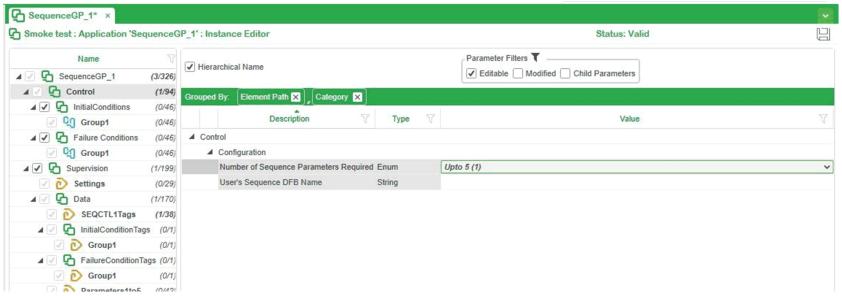
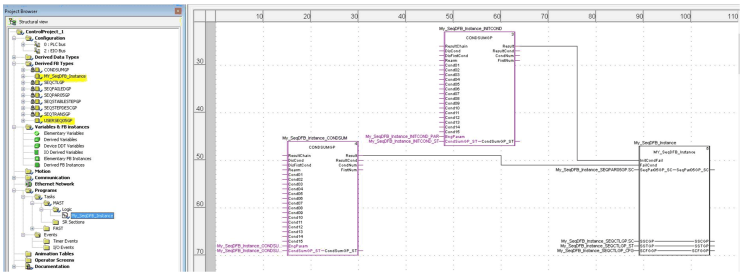
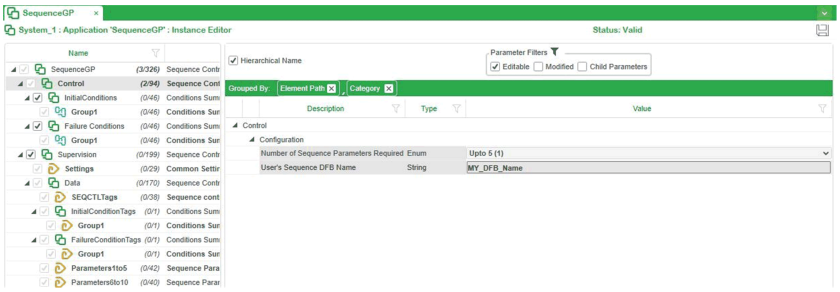
Software	Implemented as:
Automatic	It is the normal running mode. The sequence runs automatically based on the transitions defined in the controller.
Semiautomatic	In this operating mode, after the conditions defined in the transitions are met, you need to confirm the move to the next step.
Manual	The sequence does not automatically advance to a step; user can manually change from one sequence step to another.Once the sequence step is selected, operating mode will change to semi-automatic and starts the sequence step selected in manual mode.

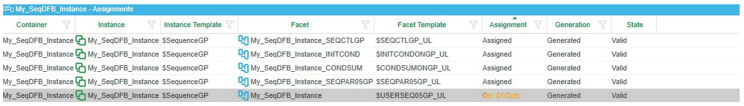
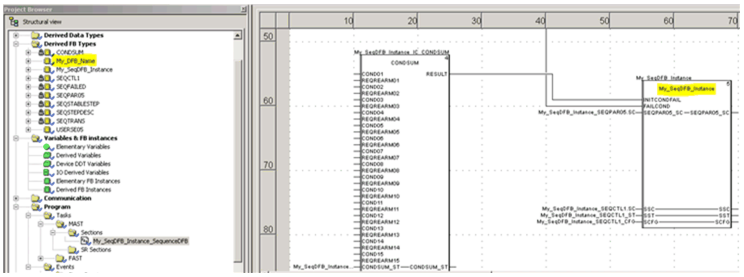
Using the $SequenceDFB Template

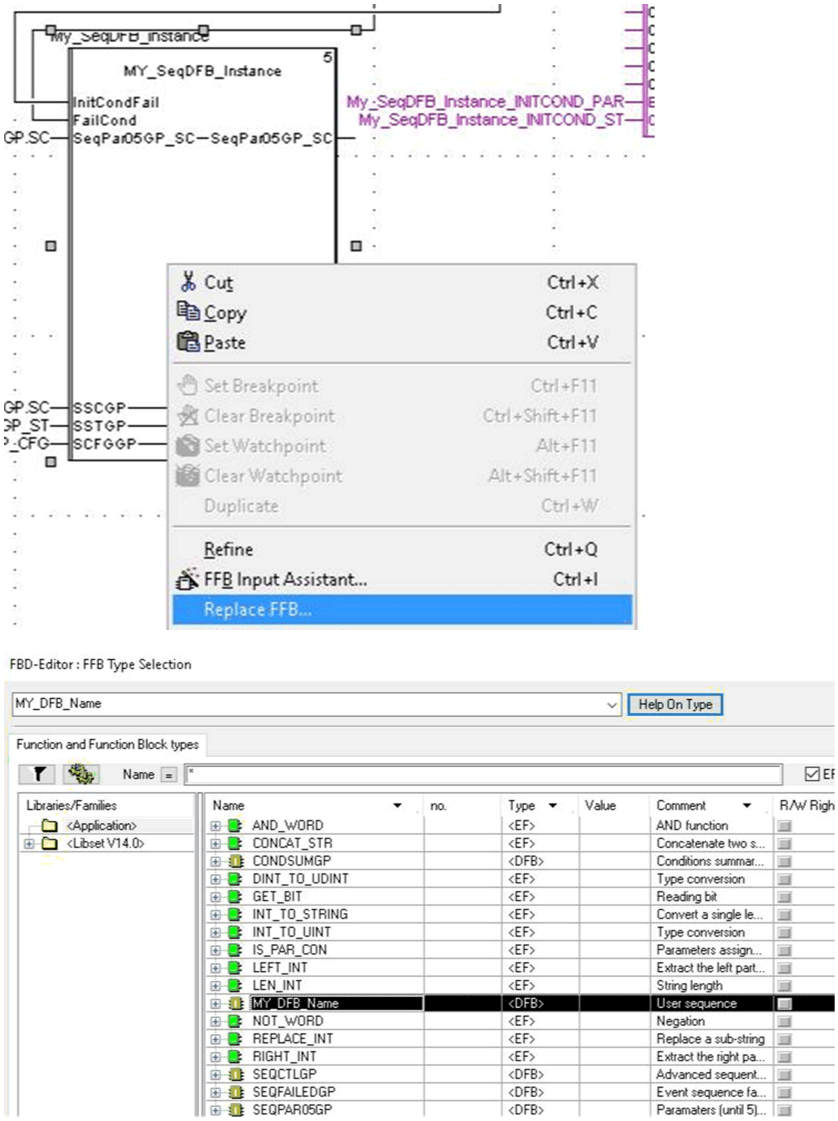
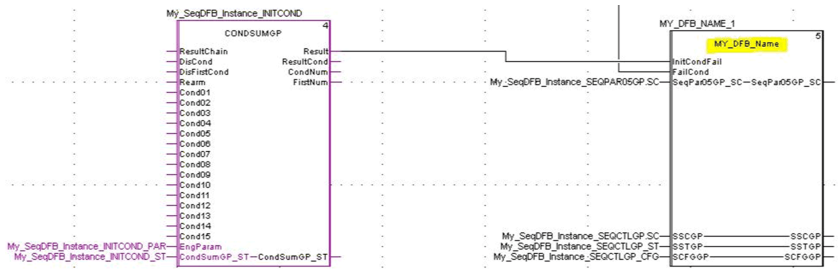
Overview

This topic explains how to use the *$SequenceGP* template instance with the help of two examples.

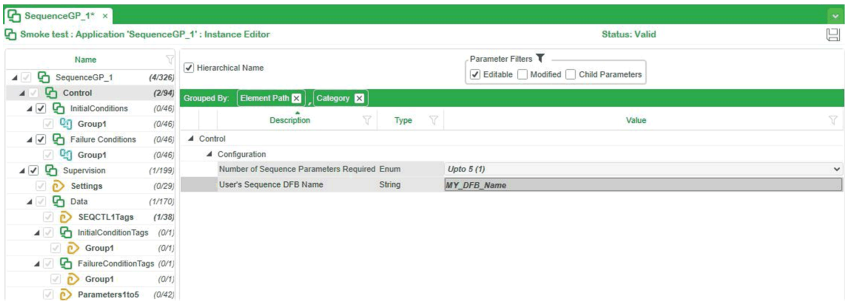
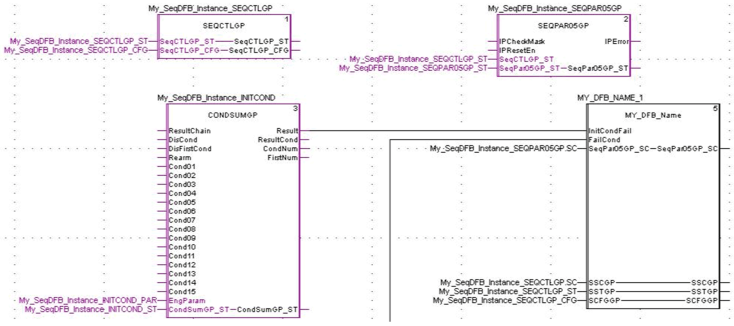
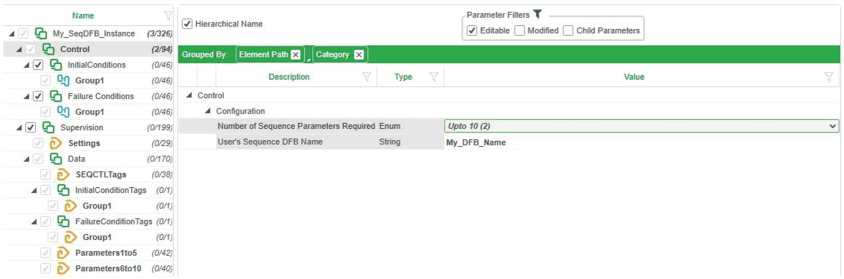
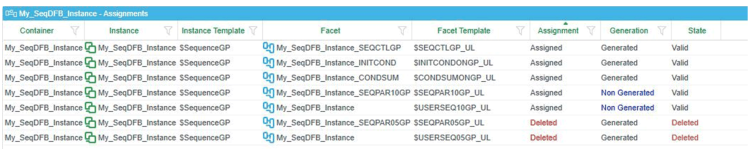
Example 1: Instantiation with Default Configuration and Subsequent Customization of the Type Name

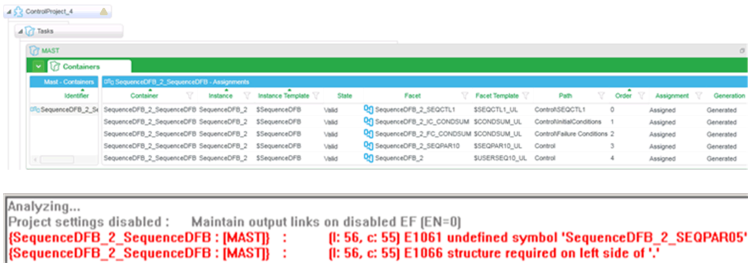
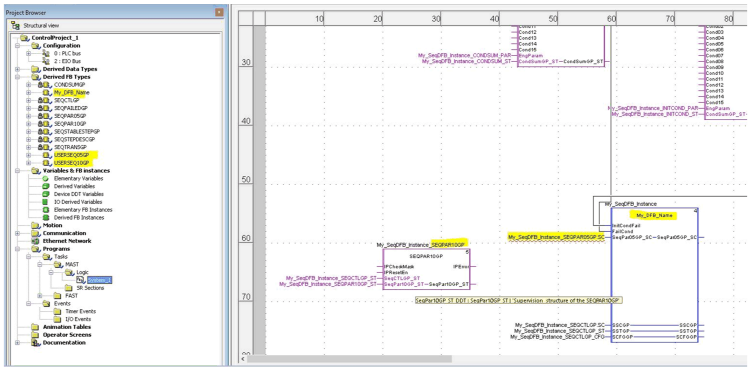
Step	Action	
1	Instance identifier: *My_SeqDFB_Instance*. Default type name (the User's *SequenceGP* Name parameter is left blank) and default number of parameters. 	
2	Assign facets of *My_SeqDFB_Instance*. 	
3	Generate the project.	
	a	The default sequence DFB type USERSEQ05GP is created but not used.
	b	A clone of the USERSE05 type is created, which has the identifier of the instance as name (*My_SeqDFB_Instance*) and an instance of this DFB type is created. 
4	Modify the type name in the Instance Editor. 	
5	Impact on assignments.	
	a	The facet that contains the USERSEQ05GP DFB type goes to **Out Of Date**.

Step	Action	
		
6	Generate the project.	
	a	Both USERSEQ05GP and *My_SeqDFB_Instance* DFB types remain.
	b	A new type is created, which has the name you entered in the Instance Editor: *My_DFB_Name*.
	c	The DFB instance is not replaced inside the section. 

Step	Action
7	**Solution:** Replace the DFB inside the section manually to use the *My_DFB_Name* type. 
8	**Result:** 

Example 2: Instantiation with Custom Type Name and Subsequent Modification of Parameters

Step	Action	
1	Modify the type name in the Instance Editor: *My_DFB_Name*. 	
2	Assign the facets of the instance and generate the project.	
	a	The default sequence DFB type USERSEQ05GP is created but not used.
	b	A clone of the USERSE05 type is created, which has the name that you have entered in the Instance Editor (*My_DFB_Name*) and an instance of this DFB type is created. 
3	Modify the number of parameters from 5 to 10 in the Instance Editor. 	
4	Assign the new facets.	
	a	Two new facets are created, which generate code related to the new number of parameters.
	b	The two facets that generate code related to the previous number of parameters (5) are deleted. 
5	Generate the project.	
	a	The analysis of the project detects an error.

Step	Action	
		
6	Refine the project.	
	a	A new sequence DFB type that corresponds to the new number of parameters is created (USERSEQ10GP) but is not used.
	b	*My_DFB_Name* type remains and its instance is not replaced inside the section.
	c	The DFB that manages the parameters (*SEQPAR05GP*, see step 2) has been replaced by a new DFB (*SEQPAR10GP*) to manage the new number of parameters. A new variable has been created (*SequenceGP_2_SEQPAR10GP*). 
	d	However, the variable reference on *My_DFB_Name* has not been updated and still refers to the previous one (*SequenceGP_2_SEQPAR05GP.SC*), which was created to manage 5 parameters. *My_DFB_Name* will not be managed by EcoStruxure Process Expert. User has to manually change the required references to support 10 parameters.
7	**Solution:**	
	a	Refine the project, goto **DataEditor** window and select **DFBTypes**.
	b	Expand **inputs/outputs**. Replace *SEQPAR05GP_SC* with *SEQPAR10GP_SC* and datatype *SEQPAR05GP_SC_DDT* with *SEQPAR10GP_SC_DDT*. My_DFB_Name | | <DFB> <inputs> <outputs> <inputs/outputs> SEQPAR10_SC | 3 | SEQPAR10_SC_DDT SSC | 18 | SEQCTL1_SC_DDT SST | 19 | SEQCTL1_ST_DDT SCFG | 20 | SEQCTL1_CFG_DDT
	c	Analyze the DFB.
	d	Find the *MY_DFB_Name* with the above changes under **section**. Edit variable to *SequenceGP_2_SEQPAR10GP.SC*.
	e	Build the project.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

SEQCTLGP - Sequential Control

What's in This Chapter

Template 662
Control - SEQCTLGP 669
Supervision 674

Overview

This chapter describes the components that provide the functionality of template, control services, and Supervision functions of the SEQCTLGP DFB.

Template

Overview

This section describes functionality, parameters, and interfaces of the $SequenceGP template.

Description

General Description

The $SequenceGP process control module template is used to monitor and manage control sequences.

Optional Services

Available optional service:

- Failure

Parameters

Parameters

The $SequenceGP template defines operation of sequence control. Alternative sequence control input facets enable the modification of the sequence control that are available.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Number of sequence parameters required	Enum	No Parameter (0)	–
	User sequence DFB name	String	–	–
Description	**InitialConditions**			

Section	Parameter description	Type	Default	Additional remarks
	Condition 1 to 15	String	–	–
	Group description		Initial conditions	–
	FailureConditions			
	Condition 1 to 15	String	–	–
	Group description		Fail-ures	–
Reset Required	**InitialConditions**			
	Condition 1 to 15	Boolean	False	–
	FailureConditions			
	Condition 1 to 15	Boolean	False	–
Disable Bypass	**InitialConditions**			
	Condition 1 to 15	Boolean	False	–
	FailureConditions			
	Condition 1 to 15	Boolean	False	–

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Operation	Normal Owner	Enum	Program (1)	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	Set Owner	Enum	Operator (1)	Privilege to change owner. This parameter is used in Operator tab, page 676.
	Set Alarm Acknowledge		Operator (1)	Privilege to acknowledge alarm. This parameter is used in Alarm tab, page 67.
	Set semi mode		Operator (1)	Privilege to change set semi mode. This parameter is used in Operator tab, page 676.
	Set manual mode			Privilege to change set manual mode. This parameter is used in Operator tab, page 676.
	Set next step			Privilege to change set next step. This parameter is used in Operator tab, page 676.
	Set scroll up			Privilege to change set scroll up. This parameter is used in Operator tab, page 676.
	Set scroll down			Privilege to change set scroll down. This parameter is used in Operator tab, page 676.
	Set start			Privilege to change set start. This parameter is used in Operator tab, page 676.
	Set hold			Privilege to change set hold. This parameter is

Section	Parameter description	Type	Default	Additional remarks
				used in Operator tab, page 676.
Security	Set restart	Enum	Operator (1)	Privilege to set restart. This parameter is used in Operator tab, page 676.
	Set stop			Privilege to change set stop. This parameter is used in Operator tab, page 676.
	Set reset			Privilege to reset. This parameter is used in Operator tab, page 676.
	Set pause			Privilege to change set pause. This parameter is used in Operator tab, page 676.
	Set time			Privilege to change set time. This parameter is used in Operator tab, page 676.

Asset and Display Parameters

For more details, refer to the topic describing the parameters of asset and display, page 43.

Data

The table describes the data parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Tag used as DATA1 to be shown on the screen	String	–	–
	Tag used as DATA2 to be shown on the screen			–
	Disable start command button	Boolean	False	–
	Disable stop command button			–
	Disable hold command button			–
	Disable pause command button			–
	Disable restart command button			–
	Disable abort command button			–
	Disable reset command button			–
	Short name for strategy 1	String	–	–
	List of applicable parameter ID's (2 to 16, separated by commas) for strategy 1 (Example: '2,3,6')			–
	Short name for strategy 2			–
	List of applicable parameter ID's (2 to 16, separated by commas) for strategy 2 (Example: '2,3,6')			–
	Short name for strategy 3			–

Section	Parameter description	Type	Default	Additional remarks
	List of applicable parameter ID's (2 to 16, separated by commas) for strategy 3 (Example: '2,3,6')			–
	Short name for strategy 4	String	–	–
	List of applicable parameter ID's (2 to 16, separated by commas) for strategy 4 (Example: '2,3,6')			–
	Short name for strategy 5			–
	List of applicable parameter ID's (2 to 16, separated by commas) for strategy 5 (Example: '2,3,6')			–
	Short name for strategy 6			–
	List of applicable parameter ID's (2 to 16, separated by commas) for strategy 6 (Example: '2,3,6')			–
	Short name for strategy 7			–
	List of applicable parameter ID's (2 to 16, separated by commas) for strategy 7 (Example: '2,3,6')			–
	Short name for strategy 8			–
	List of applicable parameter ID's (2 to 16, separated by commas) for strategy 8 (Example: '2,3,6')			–
Alarm	Held for error description	String	@(Held For Error)	–
	Held for error group		Failure	–
	Held for error severity	Enum	Low (4000)	–
Operation	Normal state	Enum	Idle & Running (7)	–
Event	Enable log	Boolean	False	Enable event logging for asset status.
Historize	Enable sequence command	Boolean	False	–
	Enable current sequence state			–
	Enable current step description			–
	Enable elapsed time			–
	Enable step description			–
	Enable new sequential step			–
	Enable transition description			–
	Enable equipment state			–
	Enable sequence emergency mode			–
	InitalConditionTags and F**ailure Conditions Tags**			
	Condition	Boolean	False	–
Input Parameter Configuration	**Parameters1to5**			
	Input parameter 01 discription	String	–	–

Section	Parameter description	Type	Default	Additional remarks
	Input parameter 01 format		####.##	–
	Input parameter 02 discription		–	–
	Input parameter 02 format		####.##	–
	Input parameter 03 discription		–	–
	Input parameter 03 format		####.##	–
	Input parameter 04 discription		–	–
	Input parameter 04 format		####.##	–
	Input parameter 05 discription		–	–
	Input parameter 05 format		####.##	–
	Parameters6to10			
	Input parameter 06 discription	String	–	–
	Input parameter 06 format		####.##	–
	Input parameter 07 discription		–	–
	Input parameter 07 format		####.##	–
	Input parameter 08 discription		–	–
	Input parameter 08 format		####.##	–
	Input parameter 09 discription		–	–
	Input parameter 09 format		####.##	–
	Input parameter 10 discription		–	–
	Input parameter 10 format		####.##	–
	Parameters11to16			
	Input parameter 11 discription	String	–	–
	Input parameter 11 format		####.##	–
	Input parameter 12 discription		–	–
	Input parameter 12 format		####.##	–
	Input parameter 13 discription		–	–
	Input parameter 13 format		####.##	–
	Input parameter 14 discription		–	–
	Input parameter 14 format		####.##	–
	Input parameter 15 discription		–	–
	Input parameter 15 format		####.##	–
	Input parameter 16 discription		–	–
	Input parameter 16 format		####.##	–

Section	Parameter description	Type	Default	Additional remarks
Output Parameter Configuration	**Parameters1to5**			
	Output parameter 01 discription	String	–	–
	Output parameter 01 format		####.##	–
	Output parameter 02 discription		–	–
	Output parameter 02 format		####.##	–
	Output parameter 03 discription		–	–
	Output parameter 03 format		####.##	–
	Output parameter 04 discription		–	–
	Output parameter 04 format		####.##	–
	Output parameter 05 discription		–	–
	Output parameter 05 format		####.##	–
	Parameters6to10			
	Output parameter 06 discription	String	–	–
	Output parameter 06 format		####.##	–
	Output parameter 07 discription		–	–
	Output parameter 07 format		####.##	–
	Output parameter 08 discription		–	–
	Output parameter 08 format		####.##	–
	Output parameter 09 discription		–	–
	Output parameter 09 format		####.##	–
	Output parameter 10 discription		–	–
	Output parameter 10 format		####.##	–
Output Parameter Configuration	**Parameters11to16**			
	Output parameter 11 discription	String	–	–
	Output parameter 11 format		####.##	–
	Output parameter 12 discription		–	–
	Output parameter 12 format		####.##	–
	Output parameter 13 discription		–	–
	Output parameter 13 format		####.##	–
	Output parameter 14 discription		–	–
	Output parameter 14 format		####.##	–
	Output parameter 15 discription		–	–
	Output parameter 15 format		####.##	–
	Output parameter 16 discription		–	–

Section	Parameter description	Type	Default	Additional remarks
	Output parameter 16 format		####.##	–
Report Parameter Configuration	**Parameters1to5**			
	Report parameter 01 to 05 format	String	–	–
	Parameters6to10			
	Report parameter 06 to 10 format	String	–	–
	Parameters11to16			
	Report parameter 11 to 16 format	String	–	–
Historize	**Parameters1to5**			
	Enable status word	Boolean	False	–
	Enable configuration word			–
	Enable input parameter 06 to 10			–
	Enable output parameter 06 to 10			–
	Enable report output parameter 11 to 16			–
	Parameters6to10			
	Enable input parameter 01 to 05	Boolean	False	–
	Enable output parameter 01 to 05			–
	Enable report output parameter 01 to 05			–
	Parameters11to16			
	Enable input parameter 11 to 16	Boolean	False	–
	Enable output parameter 11 to 16			–
	Enable report output parameter 11 to 16			–

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$SequenceGP` (also applicable for `$SequenceCE`) template as it appears in `Links Editor` and an example of other templates connected to it:

For interface identifiers, refer to the control module template interfaces, page 49.

Control - SEQCTLGP

Overview

This chapter describes the SEQCTLGP DFB.

Description

Initialization

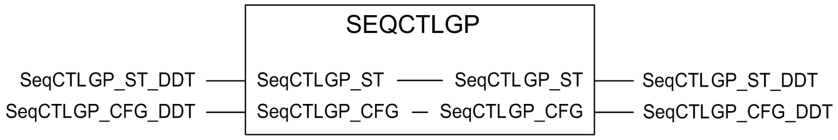
If the sequence status is unknown, it is initialized as Idle regardless of whether a controller has restarted or not. This initialization is done automatically.

DFB Representation

Representation

The DFB that is used in the program has the following aspect at the section level when imported.You can use it in any of the programming languages, although it is designed for use with the FBD language.

The figure shows an example of the SEQCTLGP DFB instance:



Implement the sequence logic that controls the component in ST language to take advantage of the sequence control potentials making the sequence reusable. That is, you can run simultaneously several instances of the sequence that you have implemented. This reduces engineering and maintenance efforts when it is necessary to set up the same control sequences in several process units.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
SeqCTLGP_ST	SeqCTLGP_ST_DDT	Provides the necessary data for monitoring and controlling sequence execution.
SeqCTLGP_CFG	SeqCTLGP_CFG_DDT	Provides the necessary data for background monitoring.

SeqCTLGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
State	–	–	UINT	Provides the status of the sequence. Read-only access to the data is contained in this bits word. Refer to the Definition of States and Commands section, page 653.
Command	–	–	UINT	Enables commands to be sent to the sequence usually from the Supervision system. Write access to the data is contained in this bits word. The DFB sets the command to 0 after processed (successfully or not). Refer to the Definition of States and Commands section, page 653.

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Sequence status word. In this word, the SEQTRANSGP auxiliary DFB is used to load the status of the transitions (maximum of 6) from the current step of the sequence to the steps in the next level of the sequence, where 0 means that the conditions in the transition are not fulfilled and 1 means they are fulfilled.
	Status1Transition	*Bit0*	BOOL	Status of the first transition
	Status2Transition	*Bit1*	BOOL	Status of the second transition
	...	...	BOOL	...
	Status6Transition	*Bit5*	BOOL	Status of the sixth transition
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
	Owner	*Bit0*	BOOL	Read/write access Enables to configure whether the sequence commands will come from the program (0) or the operator (1).
	Semi	*Bit1*	BOOL	Read/write access Enables to configure the sequence in automatic (0) or semi-automatic (1) mode.
	Manual	*Bit2*	BOOL	Read/write access Enables to configure the sequence in automatic/semi-automatic (0) or manual (1) mode. The meaning of the 1 and 2 bits combined is shown in the following table:

Bit 2	Bit 1	Mode
OFF	OFF	auto
OFF	ON	semi-automatic
ON	–	manual

	NextStep	*Bit3*	BOOL	Read/write access Enables to confirm the change of steps to the next sequence step in Semi mode. The DFB sets the signal to 0 after it is processed.
	ScrollUp	*Bit5*	BOOL	Read/write access Enables to show the steps previous to the one being shown on the current window in manual mode. The DFB sets the signal to 0 after it is processed.
	ScrollDown	*Bit6*	BOOL	Read/write access Enables to show the steps following the one being shown on the current window in manual mode. The DFB sets the signal to 0 after it is processed.
ETime	–	–	UDINT	The DFB calculates the execution time elapsed in tenths of a second for the current step of the control sequence. If execution of the program is stopped and resumed, the accumulated execution time is considered but not the time that the program was stopped.
CStepD	–	–	String [22]	Current step description (three characters for step number concatenated with seventeen characters for description).

SeqCTLGP_CFG_DDT Type

Name	Type	Description
NStep	INT	Number of the new step at which the subsequence needs to be positioned in manual mode. The DFB sets this variable to 0 after it is processed.
StepD	STRING[122]	Description of steps. Limited to 6 steps. In automatic and semi-automatic mode, it shows the destination steps specified in the current step transitions. In manual mode, it shows the list of the steps defined in the sequence. This variable format is: 6 steps, 20 characters each. Of the 20 characters of each step, the first 3 are for the step number and the next 17 are for its description. The value of this variable is assigned for the auxiliary functions.
TransD	STRING[122]	Description of transitions. Limited to 6 transitions. This variable format is 6 transitions, 20 characters each. The 20 characters are for its description. The value of this variable is assigned for the auxiliary functions.

Public Variables

Public Variable Description

Variable	Type	Description
SC	SeqCTLGP_SC_DDT	Provides the frequently needed data to monitor the sequence status and control it from the sequential control.

SeqCTLGP_SC_DDT Type

Name	Type	Description
State	UNIT	Read-only access Refer to the SeqCTLGP_ST.State input/output pin, page 669.
CStep	INT	Read-only access Number of the active sequence current step (Running, Pausing, Holding, Restarting, Stopping, or Aborting). The value 0 corresponds to the initial step of each subsequence.
Owner	BOOL	Read/write access Refer to Owner input/output pin, page 669.
Start	BOOL	Read/write access 1 = Sends a start sequence (**Start**) command if the owner is the program (SC.Owner is 0). 0 = The signal is automatically set to 0 by the DFB after the signal is processed.
Stop	BOOL	Read/write access 1 = Sends a stop sequence (**Stop**) command if the owner is the program (SC.Owner is 0). 0 = The signal is automatically set to 0 by the DFB after the signal is processed.
Hold	BOOL	Read/write access 1 = Sends a hold sequence (**Hold**) command if the owner is the program (SC.Owner is 0). 0 = The signal is automatically set to 0 by the DFB after the signal is processed.
Pause	BOOL	Read/write access 1 = Sends a pause sequence (**Pause**) command if the owner is the program (SC.Owner is 0). 0 = The signal is automatically set to 0 by the DFB after the signal is processed.
Restart	BOOL	Read/write access 1 = Sends a command to continue the sequence (**Hold**) if the owner is the program (SC.Owner is 0). 0 = The signal is automatically set to 0 by the DFB after the signal is processed.

Name	Type	Description
Abort	BOOL	Read/write access 1 = Sends a command to abort the sequence (**Abort**) if the owner is the program (SC.Owner is 0). 0 = The signal is automatically set to 0 by the DFB after the signal is processed.
Reset	BOOL	Read/write access 1 = Sends a command to reset the sequence to be able to start it (**Reset**) if the owner is the program (SC.Owner is 0). 0 = The signal is automatically set to 0 by the DFB after the signal is processed.
Resume	BOOL	Read/write access 1 = Sends a command to resume the sequence execution (**Resume**) if the owner is the program (SC.Owner is 0). 0 = The signal is automatically set to 0 by the DFB after the signal is processed.
OnEntry	BOOL	Read-only access 1 = Reports that a new sequence step has just been started. You can use this event in the logic of the sequence step actions to run any actions. The actions run only when the step starts.
OnExit	BOOL	Read-only access 1 = Reports that the execution of the sequence current step is about to finish. You can use this event in the logic of the sequence step actions to run any actions. The actions run only when the step ends.
StableStep	BOOL	Read/write access 1 = Sends a command to mark the current step as a steady step, that is: • the step with which the sequence will resume after its is held or • the step on which the sequence will be held after the sequence is paused. 0 = The signal is automatically set to 0 by the DFB after the signal is processed. As an alternative, forcing the value of this variable to 1 by using the SEQSTABLESTEPGP auxiliary DFB is recommended.
Completed	BOOL	Read/write access 1 = Sends a command to finish the current subsequence (Running, Holding, Restarting, Stopping, and so on). 0 = The signal is automatically set to 0 by the DFB after the signal is processed. As an alternative, forcing the value of this variable to 1 by using the SEQTRANSGP auxiliary DFB is recommended.
Fail	BOOL	Read/write access 1 = Sends a command to stop the sequence normal execution. 0 = The signal is automatically set to 0 by the DFB after the signal is processed. With the activation of this signal, the Holding For Detected Error subsequence is run next if the Running, Pausing, Restarting, or Paused subsequence was running previously.
NextStep	INT	Read/write access Enables the number of the next step to be run in the sequence to be set. The signal is automatically set to 0 by the DFB after the signal is processed. As an alternative, forcing the value of this variable to 1 by using the SEQTRANSGP auxiliary DFB is recommended.
Mode	INT	Read-only access Mode is encoded as: **Value** — **Mode** 0 — Automatic 1 — Semi-automatic 2 — Manual
Dest	ARRAY[1..6] of INT	Read/write access

Name	Type	Description
(Dest[1] ... Dest[6])		Defines the possible destination steps for the current step; it is limited to 6 transitions. As an alternative, forcing the value of this variable by using the SEQTRANSGP auxiliary DFB is recommended. The signals are automatically set to 0 by the DFB after the signal is processed.
DestW	WORD	Read/write access Allows the definition of the state of up to 6 transitions (bits 0 to 5) that originate with the current step. As an alternative, forcing the value of this variable by using the SEQTRANSGP auxiliary DFB is recommended. The signals are automatically set to 0 by the DFB after the signal is processed.
ETime	BOOL	Read-only access Refer to ETime input/output pin, page 669.
RunningX	BOOL	Read-only access 1 = Indicates that the current state of the sequence is Running.
HoldingX	BOOL	Read-only access 1 = Indicates that the current state of the sequence is Holding or Holding for Detected Error.
CurrentItem	BOOL	Read-only access 1 = Indicates the current position of the list of steps in Manual mode. The value –1 indicates that there has been a mode change and -2 indicates that the mode change has been identified.
NewCycle	BOOL	Read-only access 1 = Indicates that a new sequence state machine execution cycle has started.
UpdateStepD	BOOL	Read-only access 1= Internal mark indicates the need for updating the steps descriptions.
The following elements (status) are described in the Status Controller section, page 656. The Definition of Statuses and Commands, page 653 shows the sequence finite state controller.		
Idle	BOOL	Read-only access 1 = Indicates that the current state of the sequence is Idle.
Running	BOOL	Read-only access 1 = Indicates that the current state of the sequence is Running.
Complete	BOOL	Read-only access 1 = Indicates that the current state of the sequence is Complete.
Pausing	BOOL	Read-only access 1 = Indicates that the current state of the sequence is Pausing.
Paused	BOOL	Read-only access 1 = Indicates that the current state of the sequence is Paused.
Holding	BOOL	Read-only access 1 = Indicates that the current state of the sequence is Holding.
HoldingForEr-ror	BOOL	Read-only access 1 = Indicates that the current state of the sequence is Holding or Holding for Detected Error.
Held	BOOL	Read-only access 1 = Indicates that the current state of the sequence is Held.
HeldForError	BOOL	Read-only access 1 = Indicates that the current state of the sequence is Held for detected error.
Restarting	BOOL	Read-only access 1 = Indicates that the current state of the sequence is Restarting.
Stopping	BOOL	Read-only access 1 = Indicates that the current state of the sequence is Stopping.

Name	Type	Description
Stopped	BOOL	Read-only access 1 = Indicates that the current state of the sequence is Stopped.
Aborting	BOOL	Read-only access 1 = Indicates that the current state of the sequence is Aborting.
Aborted	BOOL	Read-only access 1 = Indicates that the current state of the sequence is Aborted.

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the sequential control.

Genies

Genie Properties

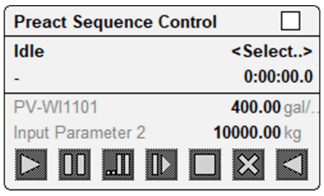
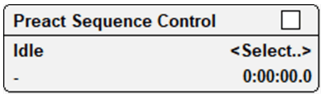
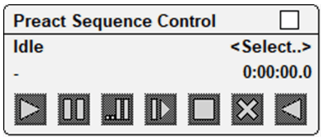
Refer to Genie Properties, page 62

Genie Icons

Refer to Genie Icons, page 63

Representation

The table describes the genies of the `sequential control` library:

Graphic symbol	Genie name	Description
	*Pannelall*	The genies display: • The current state of the sequence. • The step being executed. • The strategy that is applied (if configured and selected). • The time elapsed for the current step. • Two input Parameters • Control buttons Refer to Features, page 675 for additional information.
	*Pannelstate*	The genies display: • The current state of the sequence. • The step being executed. • The strategy that is applied (if configured and selected). • The time elapsed for the current step.
	*Pannelstatewith-buttons*	The genies display: • The current state of the sequence. • The step being executed. • The strategy that is applied (if configured and selected). • The time elapsed for the current step. • Control buttons

Features

The table describes which features each genie supports:

Features	*Pannelall*	*Pannelstate*	*Pannelstatewith-buttons*
Manages control sequences with up to eight control strategies	Yes Strategy name and applicable parameters can be provided(1).		
Up to two parameters are displayed	Yes	No	No
Buttons can be hidden	Yes	No	Yes
(1) If you leave all text fields *S1* to *S8* blank, the strategy display mechanism is disabled and parameter 1 becomes a regular parameter.			

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

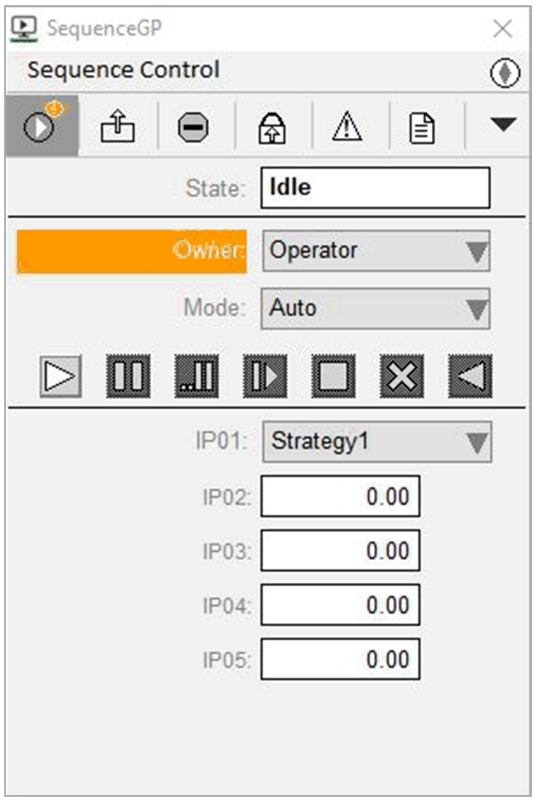
Available Tabs

During operation, clicking an on/off motor genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
- Tabs for optional functions, which appear only if configured:
  - Interlock, page 87
  - Failure, page 78

Operator Tab

The figure shows an operator tab:



Items

Overview

This section describes the variables, equipstatus and equipstate variable, abnormal and action required conditions, alarm items, events, and trend item of `sequential control`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`Transition {x}`	{y} transition status	Done/Pending	*Bit {z}*
NOTE: • {x} represents the group, range from 1 to 6.			

Item name	Description	Enumeration	Address
• {y} represents the condition, range from 1 to 6. • {z} represents the condition, range from 1 to 5.			
Bit items derived from StatusWord for the SequenceDFBParameters.			
NewData	New data available in the report parameter	On/Off	*Bit0*
IPError	Input parameters not configured		*Bit1*
Bit items derived from ConfigurationWord.			
OwnerSelect	Owner of control module	Operator/Program	*Bit0*
SemiAutomatic	Semi automatic mode command	–	*Bit1*
Manual	Manual mode command		*Bit2*
NextStep	Next step command		*Bit3*
ScrollUp	Scroll up command		*Bit5*
ScrollDown	Scroll down command		*Bit6*
Bit items derived from ConfiguratinoWord for the SequenceDFBParameters.			
ReportAck	Report parameter captured acknowledgment	On/Off	*Bit0*
IO Items			
StatusWord	Status word	–	SEQCTLGP_ST.STW
Configuration-Word	Configuration word		SEQCTLGP_ST.CFGW
Calculated Items			
EquipState	Current state of the sequential control. For more details, refer to EquipState Calculated Variable, page 677.	–	SequenceGP_EquipState
Owner	Current owner of the sequential control For more details, refer to calculated variable, page 58.		GPL_OwnerBasic
Abnormal	Shows abnormal condition of the sequential control For more details, refer to abnormal conditions, page 678.		GPL_Abnormal
ActionRequired	Shows action required condition of the sequential control For more details, refer to Action Required Conditions, page 678.		GPL_ActionRequired
Failure	Refer to Failure Variable Items, page 61.		
Interlock	Refer to Interlock Variable Items, page 60.		

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	State Value	Enumeration
Active	*302*	Running
Passive	*303*	Idle
	301	Complete
	315	Paused
	317	Held
	327	Held For Error
	313	Stopped
	311	Aborted
Active to Passive or Transistioning Active	*318*	Restarting
Passive to Active or Transistioning Passive	*314*	Pausing
	316	Holding
	326	Holding For Error
	312	Stopping
	310	Aborting

Abnormal Conditions

The table describes the abnormal conditions of sequential control:

Item name	Description	Address
`AbnormalOwner`	Abnormal owner status appears if the current owner is not a normal owner which configured at the instance level.	*Bit0*
`AbnormalMode`	Abnormal mode status appears if the current operating mode is not normal mode.	*Bit1*
`AbnormalOperatorTab`	Abnormal status of operator tab. This condition appears if the Interlock bypass is enabled.	*Bit8*
`AbnormalFailTab`	Abnormal status of failure tab.	*Bit10*
`AbnormalInterlockTab`	Abnormal status of interlock tab.	*Bit11*
`AbnormalSequenceTab`	Abnormal status of sequence tab.	*Bit13*

Action Required Conditions

The table describes the action required conditions of sequential control:

Item name	Description	Address
`ActionOperatorTab`	Action required status of operator tab.	*Bit8*
`ActionFailTab`	Action required status of failure tab.	*Bit10*
`ActionInterlockTab`	Action required status of interlock tab.	*Bit11*
`ActionSequenceTab`	Action required status of sequence tab.	*Bit13*

Alarms

The table describes the advanced alarm items that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expression
SeqAlarm	Held for error	*(PREACT_D10_State = 326)* OR *(PREACT_D10_State = 327)*

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
StateAlarm	All changes in the sequential control status and output.
ModeAlarm	All changes in the mode and owner.
ActionAlarm	All actions by the operator.

Items

No trend item is used by Supervision components for sequential control.

SEQCTLGP - Example of Use

What's in This Chapter

Description 680
DFB Representation 681
Reusable Sequence Logic 681

Overview

This chapter describes the example use of the SEQCTLGP DFB.

Description

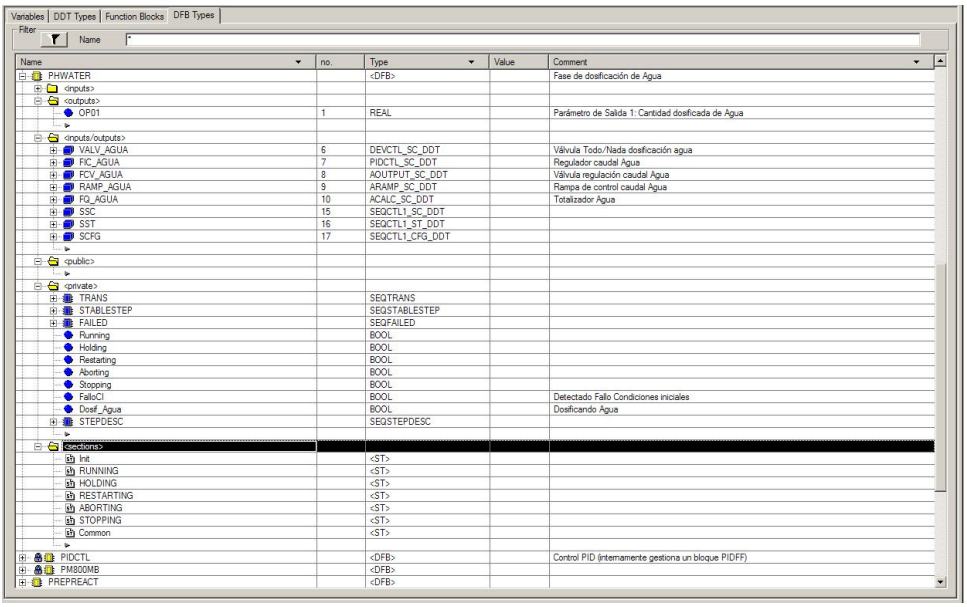
General

This chapter gives an example of an advanced control sequence controlled by the SEQCTLGP, SEQTRANSGP, SEQSTABLESTEPGP, SEQFAILEDGP, and SEQSTEPDESCGP DFBs.

The objective of the sequence is to control the loading of water in a process unit. For this purpose, a reusable component (DFB) is created so that you can use this sequence in different process units by only changing the connections of the component pins.

Reusable Sequence Definition

The reusable DFB of the user, PHWATER, declares the SEQCTLGP instance variable to manage the state of the sequence.



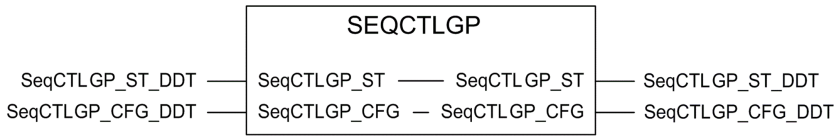
Name	no.	Type	Value	Comment
PHWATER		<DFB>		Fase de dosificación de Agua
<inputs>				
<outputs>				
OP01	1	REAL		Parámetro de Salida 1: Cantidad dosificada de Agua
<inouts/outputs>				
VALV_AGUA	6	DEVCTL_SC_DDT		Válvula Todo/Nada dosificación agua
FIC_AGUA	7	PIDCTL_SC_DDT		Regulador caudal Agua
FCV_AGUA	8	AOUTPUT_SC_DDT		Válvula regulación caudal Agua
RAMP_AGUA	9	ARAMP_SC_DDT		Rampa de control caudal Agua
FQ_AGUA	10	ACALC_SC_DDT		Totalizador Agua
SSC	15	SEQCTL1_SC_DDT		
SST	16	SEQCTL1_ST_DDT		
SCFG	17	SEQCTL1_CFG_DDT		
<public>				
<private>				
TRANS		SEQTRANS		
STABLESTEP		SEQSTABLESTEP		
FAILED		SEQFAILED		
Running		BOOL		
Holding		BOOL		
Restarting		BOOL		
Aborting		BOOL		
Stopping		BOOL		
FalloCI		BOOL		Detectado Fallo Condiciones iniciales
Dosif_Agua		BOOL		Dosificando Agua
STEPDESC		SEQSTEPDESC		
<sections>				
Init		<ST>		
RUNNING		<ST>		
HOLDING		<ST>		
RESTARTING		<ST>		
ABORTING		<ST>		
STOPPING		<ST>		
Common		<ST>		
PIDCTL		<DFB>		Control PID (internamente gestiona un bloque PIDFF)
PM800MB		<DFB>		
PREPREACT		<DFB>		

NOTE: Declaration of the auxiliary components manages the transitions (SEQTRANSGP), steady-step flags (SEQSTABLESTEPGP), and notifications to trigger the holding subsequence (SEQFAILEDGP).

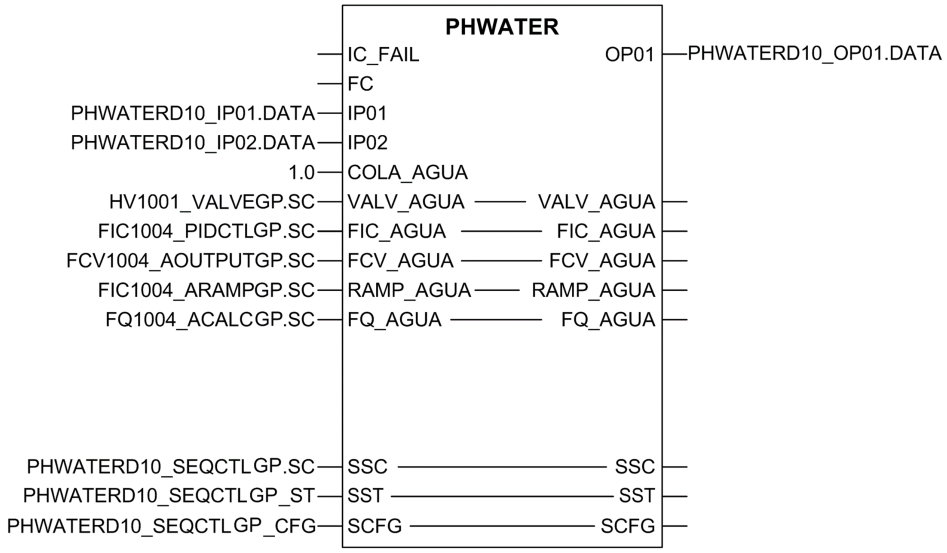
DFB Representation

Representation

The following diagrams include the calls to the sequence created by the user as well as to the control block that will control it (SEQCTLGP).



NOTE: The call to the block controls the execution of the advanced control sequence.



NOTE: The call to the functional block implements the control sequence of the user and will reuse for different process units if needed.

Reusable Sequence Logic

Sequence Logic With Description

The basic structure of the sequence as well as the steps that are defined for the sequence objective are included: water dosing. The parts of the code that make up the basic structure of the sequence are marked in bold.

(Initialization)

The following common actions have to be executed before the sequence execution:

```
Running := SSC.RUNNINGX;

Holding := SSC.HOLDINGX;

Restarting := SSC.RESTARTINGX;

Aborting := SSC.ABORTINGX;

Stopping := SSC.STOPPINGX;
```

NOTE: The default actions to be taken by DFB are defined in this section. However, if any section has to be added by user and actions need to be taken before execution of that section, it has to be added here.

(Running or Pausing Subsequence (302/314))

```
STEPDESC (1, 'Init', SSC, SST, SCFG);
STEPDESC (2, 'Valves ON', SSC, SST, SCFG);
STEPDESC (3, 'Wait Valves ON', SSC, SST, SCFG);
STEPDESC (4, 'Dosing Water', SSC, SST, SCFG);
STEPDESC (5, 'Valves OFF', SSC, SST, SCFG);
STEPDESC (6, 'Waiting for queue', SSC, SST, SCFG);
```

NOTE: Definition of the descriptions for the subsequence steps.

(Steps and Transitions)

```
case SSC.CSTEP of
```

1: (#RunningStep0# Initial Step)

NOTE: First step with a value of 1. The allowed values for steps are 1 to 999.

(Check initial conditions)

```
if IC_FAIL then
FalloCI := true;
```

`FAILED (SSC);` (Initial conditions not satisfied)

NOTE: Report detected failure to trigger the holding subsequence.

```
else
FalloCI := false;
```

(Initializations)

```
OP01:=0.0;
```

(Reset and water totalizer activation)

```
FQ_AGUA.TOTALRST:=true;
FQ_AGUA.TOTALEN:=true;
```

`TRANS (true, 'True', SSC.CSTEP+1, SSC, SST, SCFG);` (Next Step)

NOTE: Unconditional transition to the next step.

```
end_if;
```

2: (Start Water dosing)

```
if SSC.ONENTRY then
```

`STABLESTEP (SSC);` (In this step, execution would restart in the event of Hold/Continue)

NOTE: Execute the actions only for the first time.

NOTE: Flag this step as steady.

```
DOSIF_AGUA:=true;
VALV_AGUA.REM:=false;
VALV_AGUA.LSP := true;
FCV_AGUA.REM:=true;
FIC_AGUA.REM:=false;
FIC_AGUA.MAN:=true;
```

`FIC_AGUA.LOP:=10.0;` (Initial valve opening)

```
end_if;
```

`TRANS (true, 'trans1', SSC.CSTEP+1, SSC, SST, SCFG);` (Next step)

`TRANS (false, 'trans2', SSC.CSTEP+2, SSC, SST, SCFG);` (Next step)

`TRANS (false, 'trans3', SSC.CSTEP+3, SSC, SST, SCFG);` (Next step)

`TRANS (false, 'trans4', SSC.CSTEP+4, SSC, SST, SCFG);` (Next step)

`TRANS (false, 'trans5', SSC.CSTEP+5, SSC, SST, SCFG);` (Next step)

`TRANS (false, 'trans6', SSC.CSTEP+6, SSC, SST, SCFG);` (Next step)

NOTE: The evaluation priority for these transitions is the order in which they are shown (up down; these transitions are not functional, and have only been included as an example in this case). You can define a maximum of 6 transitions.

3: (Wait for open valve)

`TRANS (VALV_AGUA.HIGHPOS, 'Valve open?', SSC.CSTEP+1, SSC, SST, SCFG);` (Next step)

4: (Wait for end of Water dosing)

```
if SSC.ONENTRY then
```

`RAMP_AGUA.REM := false;` (Setpoint flow ramp)

`RAMP_AGUA.LTARGETSP:=IP02;` (Target set-point)

```
RAMP_AGUA.LSTART:=true
FIC_AGUA.REM:=true;
FIC_AGUA.AUTO:=true;
```

`FIC_AGUA.LSP:=IP02;` (Flow regulation)

```
end_if;
TRANS ((IP01 - COLA_AGUA) <= OP01, 'Dosing finished?', SSC.CSTEP + 1, SSC, SST, SCFG);
```

5: (End Water dosing)

```
if SSC.ONENTRY then
RAMP_AGUA.LSTART:=false;
FIC_AGUA.REM:=false;
FIC_AGUA.MAN:=true;
FIC_AGUA.LOP:=0.0;
VALV_AGUA.LSP:=false;
```

`STABLESTEP (SSC);` (In this step, execution will restart in the event of Hold/Continue)

```
end_if;
TRANS (VALV_AGUA.LOWPOS, 'Valve Closed?', SSC.CSTEP+1, SSC, SST, SCFG);
```

6: (Wait for Water queue)

```
if SSC.ONENTRY then
```

(Wait for `COLA_AGUA` (Water queue) to fall)

`STABLESTEPGP (SSC);` (In this step, execution will restart in the event of Hold/Continue)

```
end_if;
TRANS (SSC.ETIME >= 50, 'Waiting for queue', -1, SSC, SST, SCFG);
```
(End of sequence)

NOTE: Complete sequence (go to COMPLETE) when the transition condition is met.

```
end_case;
```

(Actions common to several steps)

(Detected failure condition monitoring after initializations)

```
if SSC.CSTEP > 0 and FC then
FAILED (SSC); (Trigger execution of the Hold sequence)
end_if;
```

(*Holding* Subsequence (316/326))

```
STEPDESC (1, 'Init', SSC, SST, SCFG);
STEPDESC (1, 'Closing Valves', SSC, SST, SCFG);
```

Steps and Transitions)

```
case SSC.CSTEP of
```

1: (Initial Step)

```
if FalloCI then
dosif_agua:=false;
end_if;
TRANS (FalloCI, 'Not InitCond', -1, SSC, SST, SCFG); (End of sequence)
TRANS (NOT FalloCI, 'InitCond OK', SSC.CSTEP+1, SSC, SST, SCFG); (Next step)
```

2:

```
if SSC.ONENTRY then
VALV_AGUA.LSP:=false;
RAMP_AGUA.LSTART:=false;
FIC_AGUA.MAN:=true;
FIC_AGUA.LOP:=0.0; (Close control valve)
end_if;
If (dosif_agua) AND SSC.ETIME >= 50 then
dosif_agua:=false;
end_if;
TRANS (SSC.ETIME >= 50 or not dosif_agua, 'Waiting for queue', -1, SSC, SST, SCFG); (End of sequence)
end_case;
```

(Restarting subsequence(318))

```
STEPDESC (1, 'Init', SSC, SST, SCFG);
```

(Steps and Transitions)

```
case SSC.CSTEP of
```

1: (Initial Step)

```
TRANS (true, 'True', -1, SSC, SST, SCFG); (End of sequence)
end_case;
```

(Aborting subsequence(310))

```
STEPDESC (1, 'Closing Valves', SSC, SST, SCFG);
```

(Steps and Transitions)

```
case SSC.CSTEP of
```

1: (Initial Step)

```
VALV_AGUA.REM:=false;
VALV_AGUA.LSP:=false;
RAMP_AGUA.LSTART:=false;
FIC_AGUA.MAN:=true;
FIC_AGUA.LOP:=0.0;
```
(Close control valve)

```
dosif_agua:=false;
TRANS (true, 'True', -1, SSC, SST, SCFG);
```
(End of sequence)

```
end_case;
```

(Stopping subsequence(312))

```
STEPDESC (1, 'Closing Valves', SSC, SST, SCFG);
STEPDESC (2, 'Waiting for queue', SSC, SST, SCFG);
```

Steps and Transitions)

```
case SSC.CSTEP of
```

1: (Initial Step)

```
VALV_AGUA.REM:=false;
VALV_AGUA.LSP:=false;
RAMP_AGUA.LSTART:=false;
FIC_AGUA.MAN:=true;
FIC_AGUA.LOP:=0.0;
```
(Close control valve)

```
TRANS (true, 'True', SSC.CSTEP+1, SSC, SST, SCFG);
```
(Next step)

2:

```
If (dosif_agua) AND SSC.ETIME >= 50 then
dosif_agua:=false;
end_if;
TRANS (SSC.ETIME >= 50 or not dosif_agua, 'Waiting for queue', -1,
SSC, SST, SCFG);
```
(End of sequence)

```
end_case;
```

Common

Common actions to be executed after the sequence execution.

NOTE: No default actions are defined in this section. However, if any action has to be taken after execution of section, it has to be added here.

Auxiliary Functions of Sequential Control

What's in This Chapter

Control - SEQTRANSGP 686
Control - SEQFAILEDGP 688
Control - SEQSTABLESTEPGP 689
Control - SEQSTEPDESCGP 690

Overview

This section describes the function blocks that supplement the SEQCTLGP, page 669 DFB and that have the objective of facilitating the programming of the control sequence.

Control - SEQTRANSGP

Overview

This section describes the SEQTRANSGP DFB.

Description

General

The main objective of SEQTRANSGP DFB is to set up the control sequence transitions based on the condition to be evaluated and the next step to be executed when this expression is fulfilled.

Additionally, the DFB manages the information necessary to graphically represent the state of the transitions and steps following the current step (DEST and DESTW fields of the SeqCTLGP_ST_DDT sequence state structure).

Include call to this function in the logic of the steps (normally in ST language) to program the transitions between the sequence steps.

Function Call

The function call is carried out from the steps of the control subsequences (Running, Restarting, Holding, and so on) normally in ST language to program the transitions between these steps.

If several SEQTRANS calls are included in the same step and if more than one of them is fulfilled in the same step execution cycle, the step corresponding to the transition that is fulfilled and was programmed first will run. The number of SEQTRANS calls from one single step is limited to 6.

```
TRANS (CONDITION := (*BOOL*),

DESC := (*string[22]*),

NEXTSTEP := (*INT*),

SeqCTLGP_SC := (*SeqCTLGP_SC_DDT*)

SeqCTLGP_ST := (*SeqCTLGP_ST_DDT*),

SeqCTLGP_CFG := (*SeqCTLGP_CFG_DDT*))
```

A function call example is included below:

4: (Wait for end of reactive dosing)

```
if SSC.ONENTRY then
BISEN2_REACT.REM:=false;
BISEN2_REACT.LSP:=true;
end_if;
BISEN2_REACT.LSPSEL:=(OP01 < (IP01 - 10.0)); (Speed 1 or 2)
TRANS ((IP01 - COLA_REACT) <= OP01, 'Dosing finished?', SSC.
CSTEP+1, SSC, SST, SCFG);
```

In the previous example, the TRANS call included in step 4 of the sequence corresponds to a `SEQTRANSGP` block instance to which the following parameters are passed:

- `(IP01 - COLA_REACT) <= OP01OS`: Corresponds to the expression of the condition that needs to be fulfilled for the transition to be executed. You can include complex expressions that result in a Boolean value (true or false).
- `Dosing finished?`: Corresponds to the description of the transition.
- `SSC.CSTEP+1`: Corresponds to the next step that needs to be executed when the transition is fulfilled. In this case, the next step (step 5) is executed after the current step (step 4). You can include any expression jumping to an existing step in the sequence.
- `SSC`: Corresponds to the `SC` public variable of the `SEQCTLGP` block instance that is controlling the sequence execution.
- `SST`: Corresponds to the `DDT_ST` variable of the `SEQCTLGP` block instance that is controlling the sequence execution.
- `SCFG`: Corresponds to the `DDT_CFG` variable of the `SEQCTLGP` block instance that is controlling the sequence execution.

Inputs

Input Parameter Description

Parameter	Type	Description
CONDITION	BOOL	1 = Indicates that the condition for executing the transition is fulfilled.
DESC	STRING[22]	Provides the description of the transition to be represented in the monitoring program. The character string is limited to 20 characters.
NEXTSTEP	INT	Subsequence step number to which a jump needs to be made when the transition is fulfilled. Refer to the CONDITION input. The permissible values for the steps go from 1 to 999 with the exception of NEXTSTEP to -1. This indicates that when the transition is fulfilled, the corresponding subsequence (to go, for example, from Running to Complete, from Holding to Held, and so on) will be completed.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
SeqCTLGP_SC	SeqCTLGP_SC_DDT	Corresponds to the SC public variable of the SEQCTLGP block instance that is controlling the sequence execution.
SeqCTLGP_ST	SeqCTLGP_ST_DDT	Provides the data necessary to communicate with the SEQCTLGP block that is controlling the sequence. Refer to the SEQCTLGP block, page 669 for more details.
SeqCTLGP_CFG	SeqCTLGP_CFG_DDT	Provides the data necessary to communicate with the SEQCTLGP block that is controlling the sequence. Refer to the SEQCTLGP block, page 669 for more details.

Control - SEQFAILEDGP

Overview

This section describes the SEQFAILEDGP DFB.

Description

General

The main objective of the SEQFAILEDGP DFB is to report detected errors to the sequence management block (SEQCTLGP) during sequence execution so that the *HoldingFE* subsequence is triggered.

Function Call

The function call is made from the Running, Restarting, Pausing, or Paused control subsequence steps normally in ST language with the purpose of reporting detected errors in the process that is being controlled. The detected error requires the hold sequence to be executed (*HoldingFE*).

A function call example is included below:

(Detected failure condition monitoring after initializations)

```
if SeqCTLGP.CSTEP > 0 and INITCONDFAIL then
```

`FAILED (SeqCtlGP);` (Trigger execution of the Hold sequence)

```
end_if;
```

In the previous example, the *FAILED* call that is included in the logic monitors the detected error conditions in each execution cycle (in this particular case, but could be associated with a specific step) corresponds to a SEQFAILEDGP DFB block instance to which the following parameters are passed:

- SeqctlGP: Corresponds to the name of the SC public variable of the SEQCTLGP block instance that is controlling the execution of the sequence.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
SeqCTLGP_SC	SeqCTLGP_SC_DDT	Provides the data necessary to communicate with the SEQCTLGP block that is controlling the sequence. Refer to the SEQCTLGP block, page 669 for more details.

Control - SEQSTABLESTEPGP

Overview

The section describes the SEQSTABLESTEPGP DFB.

Description

General

The main objective of the SEQSTABLESTEPGP DFB is to report to the sequence management block (SEQCTLGP) that the current running sequence step corresponds to a step, that is, the re-entry point in the normal execution (Running) sequence executes after restarting execution (Restarting).

Function Call

The function call is performed from the Running control subsequence steps normally in ST language for the purpose of reporting. The current step is the entry point to which the program should return when the Running subsequence is resumed after finishing the Restarting subsequence. Basically, it enables to resume the sequence execution that verifies the appropriate actions are performed or not.

For example, suppose a valve has to be opened. After the valve is opened, the next step involves turning on a pump. While waiting for the valve to open, a detected failure that makes the sequence stop occurs (Holding subsequence execution). You can resume execution by running the restart (Restarting) subsequence. After this, the program returns to the normal running subsequence (Running). After the program goes back to this Running subsequence, the intention is to make another attempt at opening the valve (because most likely, it has been closed during the Holding subsequence) not to start the pump in proceeding directly. Therefore, in the valve opening step, the step is marked and does not call the function SEQSTABLESTEPGP in the pump starting step.

The following is an example of a function call:

0: (End Water dosing)

```
if SeqCtlGP.ONENTRY then

RAMP_AGUA.LSTART:=false;

FIC_AGUA.REM:=false;

FIC_AGUA.MAN:=true;

FIC_AGUA.LOP:=0.0;

VALV_AGUA.LSP:=false;
```

```
STABLESTEPGP (SeqCtlGP);

end_if;

TRANS (VALV_AGUA.LOWPOS, SeqCTLGP.CSTEP+1, SeqCtlGP);
```

In the previous example, the STABLESTEP call included in the logic that monitors detected failure conditions in each execution cycle (in this particular case, but it could be associated with a specific step) corresponds to a SEQSTABLESTEPGP DFB block instance to which the following parameters are passed:

- SeqctlGP: Corresponds to the name of the SC public variable of the SEQCTLGP block instance that is controlling the sequence execution.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
SeqCTLGP_SC	SeqCTLGP_SC_DDT	Provides the data necessary to communicate with the SEQCTLGP block that is controlling the sequence. Refer to the SEQCTLGP block, page 669 for more details.

Control - SEQSTEPDESCGP

Overview

This section describes the SEQSTEPDESCGP DFB.

Description

General

The main objective of the SEQSTEPDESCGP DFB is to assign descriptions to the steps that are defined in the sequence to allow you to monitor them from the Supervision system.

Function Calls

The function call is made from the control subsequence steps (Running, Restarting, Holding, and so on) at the beginning of the code to provide the description of the subsequence steps for their representation in the monitoring program. It is normally written in ST language.

Include as many SEQSTEPDESC calls as there are steps defined in the subsequence or as there are steps that need to be represented in the Supervision system.

The syntax of the function is the following:

```
STEPDESC (STEPNUMBER := (INT),

DESC := (string[18]),

SeqCTLGP_SC := (SeqCTLGP_SC_DDT),

SeqCTLGP_ST := (SeqCTLGP_ST_DDT),
```

```
SeqCTLGP_CFG := (SeqCTLGP_CFG_DDT),
```

The following is an example of a function call:

(Running or Pausing Subsequence (302 / 314))

(Step descriptions)

```
STEPDESCGP (1, 'Init', SSC, SST, SCFG);
STEPDESCGP (2, 'Valves ON', SSC, SST, SCFG);
STEPDESCGP (3, 'Wait Valves ON', SSC, SST, SCFG);
STEPDESCGP (4, 'Dosing Reactive', SSC, SST, SCFG);
STEPDESCGP (5, 'Motor OFF', SSC, SST, SCFG);
STEPDESCGP (6, 'Waiting for queue', SSC, SST, SCFG);
```

(Steps and Transitions)

```
case SSC.CSTEP of
```

1: (#RunningStep0# Initial Step)

(Check initial conditions)

```
if IC_FAIL then
FalloCI := true;
```

`FAILED (SSC);` (Initial conditions not satisfied)

```
else
FalloCI := false;
```

(Initialization)

In the above example, the STEPDESC calls included at the beginning of the subsequence correspond to a `SEQSTEPDESCGP` DFB block instance to which the following parameters are passed in case of the first call:

- 1: Corresponds to step number 1.
- `Init`: Corresponds to the definition of the step.
- `SSC`: Corresponds to the `SC` public variable of the `SEQCTLGP` block instance that is controlling the sequence execution.
- `SST`: Corresponds to the `DDT_ST` variable of the `SEQCTLGP` block instance that is controlling the sequence execution.
- `SCFG`: Corresponds to the `DDT_CFG` variable of the `SEQCTLGP` block instance that is controlling the sequence execution.

Inputs

Input Parameter Description

Parameter	Type	Description
STEPNUMBER	INT	Number of the subsequence step. The values allowed for the steps are from 1 to 999.
DESC	STRING[22]	Description of the number of the subsequence step. The character string is limited to 17 characters. This is the value that will be shown in the sequence monitoring program.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
SeqCTLGP_SC	SeqCTLGP_SC_DDT	Provides the data necessary to communicate with the SEQCTLGP block that is controlling the sequence. Refer to the SEQCTLGP, page 669 block for more details.
SeqCTLGP_ST	SeqCTLGP_ST_DDT	Provides the data necessary to communicate with the SEQCTLGP block that is controlling the sequence. Refer to the SEQCTLGP, page 669 block for more details.
SeqCTLGP_CFG	SeqCTLGP_CFG_DDT	Provides the data necessary to communicate with the SEQCTLGP block that is controlling the sequence. Refer to the SEQCTLGP, page 669 block for more details.

Public Variables

Public Variable Description

Parameter	Type	Description
ORDERNUMBER	INT	Read-only public variable reserved for building the list of steps in manual mode that will be shown in the monitoring program.
MAXORDERNUMBER	INT	Read-only public variable reserved for building the list of steps in manual mode that will be shown in the monitoring program.

SEQPARxxGP - Sequence Parameter Management

What's in This Chapter

Control 693

Overview

This chapter describes the SEQPARxxGP DFB.

Control

Overview

This chapter describes the SEQPARxxGP DFB.

Description

General

The SEQPAR05GP, SEQPAR10GP and SEQPAR16GP DFBs manage up to 5, 10, and 16 sequence parameters (input, output, and reports) respectively. The DFBs facilitate the management of parameters from advanced control sequences implemented with the General Purpose Library SEQCTLGP DFB.

These DFBs manage the sequence parameters with a data structure (DDT) making it easier to implement reusable control sequences (the number of pins in a DFB is limited to 32 input pins and 32 output pins) because a single variable is used to provide access to the data that the sequence manages.

Function Description

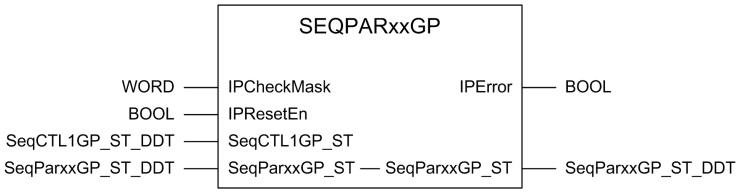
The main functions of the DFB are described in the following table:

Function	Description
Input Parameter Consistency	Optionally, the DFB initializes the values of the input parameters with a value (-0.0) that enables to determine whether or not the applicable parameters of the sequence have been correctly downloaded onto the controller. This way, the sequence can be stopped from starting with an incomplete or inconsistent set of input parameters.
Owner	The DFB manages control system level, which is the owner (Operator or Program) based on the owner of the associated sequence. As a result, it is responsible for defining the input parameters of the sequence.
Output Parameters	The DFB manages the output parameters generated from the control sequence.
Report Parameters	The DFB copies the output parameters to report parameters when the sequence ends and maintains them until the next sequence end. This facilitates the acquisition of these parameters from the monitoring subsystem to generate reports.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
IPCheckMask	WORD	Optionally, enables to connect a bits word (16) to configure applicable input parameters. Only the applicable parameters will be checked for consistency. Refer to the IPResetEn input and IPError output pin, page 694. If no data is connected to this input, the consistency check for input parameter is disabled.
		Bit — **SP** is calculated as:
		0 (2^0) — Input parameter 1 is applicable (1) or not (0).
		1 (2^1) — Input parameter 2 is applicable (1) or not (0).
		... — ...
		15 (2^{15}) — Input parameter 16 is applicable (1) or not (0).
IPResetEn	BOOL	1 = Enables to configure the input parameters need to be initialized with the -0.0 value when the associated control sequence switches to an Idle status. This configuration together with the IPCheckMask input pin is used to determine whether or not the input parameters are correctly downloaded (refer to the IPError output pin, page 694).
SeqCTLGP_ST	SeqCTLGP_ST_DDT, page 669	Enables to connect the state structure of the control sequence for which the parameters need to be managed. The DFB uses this structure to determine the state of the sequence (Idle, Running and so on) and its owner (Operator or Program).

Outputs

Output Parameter Description

Parameter	Type	Description
IPError	BOOL	1 = A detected error in the assignment. 0 = The check is not activated. The check is only carried out if the IPCheckMask input is connected. If the IPCheckMask input is connected, a check is run to verify whether the applicable parameters as defined in the connected bit mask have a value other than -0.0 or not. The responsibility of assigning the -0.0 value belongs to the DFB or the programmer depending on how the IPResetEn input is configured.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
SeqParxxGP_ST	SeqParxxGP_ST_DDT	Provides the data that is necessary for managing the types of parameters that the DFB manages input, output, and report.

SeqParxxGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	NewData	Bit0	BOOL	Indicates that new data (1) is available in the report parameters (RPTxx fields of the SeqParxxGP_ST structure). The DFB activates this signal when the control sequence switches to Complete, Stopped, or Aborted and deactivates this signal (0) when its processing is confirmed from the RPTack (SeqParxxGP_ST.CFGW.0) signal.
	IPError	Bit1	BOOL	Refer to the IPError output pin, page 694.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
	RPTack	Bit0	BOOL	Read/write access Enables to reset the NewData (SeqParxxGP_ST.STW.0) signal. This signal is automatically set to 0 by the DFB after it has been processed.
IP01	–	–	REAL	Read/write access Input parameter 1. Input parameter management: • The DFB initializes the input parameters with a value of -0.0 if the IPResetEn input is enabled when the sequence state switches to Idle. • If the owner of the control sequence is the Operator (SEQCTLGP_ST.CFGW.0 = 1), this data is copied to the IPxx fields of the SC public variable. In other words, the input parameters are received from the monitoring or batch control subsystem in the Operator mode. • If the owner of the control sequence is the Program (SEQCTLGP_ST.CFGW.0 = 0), the input parameters are overwritten with the data in the IPxx fields of the SC public variable.
...	–	–		...
IPxx	–	–	REAL	Read/write access Input parameter xx.
OP01	–	–	REAL	Read-only access Output parameter 1. Output parameter management: • The values in the OPxx fields of the SC public variable are continuously copied to these output parameters so that they can be accessed from the monitoring subsystem. Therefore, the control sequence of the user needs to define the value of the OPxx parameters of the SC public variable to indicate the results of the execution of the sequence.
...	–	–		...
OPxx	–	–	REAL	Read-only access Output parameter xx.
RPT01			REAL	Read-only access Report parameter 1. Report parameter management:

Name	Bit Name	Bit	Type	Description
				• The SeqParxxGP DFB copies the values of the output parameters to the report parameters when the execution of the associated control sequence ends (it switches to the Complete, Stopped, or Aborted state) to facilitate their acquisition from the report subsystem. Refer to the NewData (SeqParxxGP_ST.STW.0) input/output.
...				...
RPTxx			REAL	Read-only access Report parameter xx.

Public Variables

Public Variable Description

Variable	Type	Description
SC	SeqParxxGP_SC_DDT	Provides the data needed to manage the parameters of a sequence from the control sequence itself or from a different external mechanism.

SeqParxxGP_SC_DDT Type

Name	Type	Description
IPError	BOOL	Read-only access Refer to the IPError output pin, page 694.
IP01	REAL	Read/write access Input parameter 1. Input parameter management: • The DFB initializes the input parameters with a value of -0.0 if the IPResetEn input is enabled when the sequence status switches to Idle. • If the owner of the control sequence is the Operator (SEQCTLGP_ST.CFGW.0 = 1), this data is copied to the IPxx fields of the SC public variable. In other words, the input parameters are received from the monitoring or batch control subsystem in the Operator mode. • If the owner of the control sequence is the Program (SEQCTLGP_ST.CFGW.0 = 0), the input parameters are overwritten with the data in the IPxx fields of the SC public variable.
...		...
IPxx	REAL	Read/write access Input parameter xx.
OP01	REAL	Read-only access Output parameter 1. Output parameter management: • The values in the OPxx fields of the SC public variable are continuously copied to these output parameters so that you can access the parameters from the monitoring subsystem. Therefore, the control sequence of the user needs to define the value of the OPxx parameters of the SC public variable to indicate the results of the sequence execution.
...		...
OPxx	REAL	Read-only access Output parameter xx.

USERSEQxxGP - User Sequence Parameter

What's in This Chapter

Control 697

Overview

This chapter describes the USERSEQxxGP DFB.

Control

Overview

This chapter describes the USERSEQxxGP DFB.

Description

General

The USERSEQ0GP, USERSEQ5GP, USERSEQ10GP and USERSEQ16GP DFBs manage up to 0, 5, 10, and 16 sequence parameters (input, output, and reports) respectively. The DFBs facilitate the management of parameters from advanced control sequences implemented with the General Purpose Library SEQCTLGP DFB.

These DFBs manage the user sequence parameters with a data structure (DDT) making it easier to implement reusable control sequences (the number of pins in a DFB is limited to 32 input pins and 32 output pins) because a single variable is used to provide access to the data that the sequence manages.

Function Description

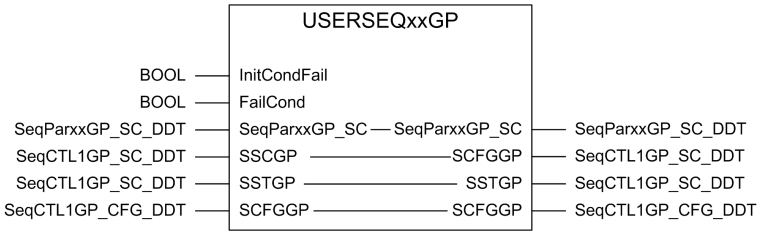
The main functions of the DFB are described in the following table:

Function	Description
Input Parameter Consistency	Optionally, the DFB initializes the values of the input parameters with a value (-0.0) that enables to determine whether or not the applicable parameters of the sequence have been correctly downloaded onto the controller. This way, the sequence can be stopped from starting with an incomplete or inconsistent set of input parameters.
Owner	The DFB manages control system level, which is the owner (Operator or Program) based on the owner of the associated sequence. As a result, it is responsible for defining the input parameters of the sequence.
Output Parameters	The DFB manages the output parameters generated from the control sequence.
Report Parameters	The DFB copies the output parameters to report parameters when the sequence ends and maintains them until the next sequence end. This facilitates the acquisition of these parameters from the monitoring subsystem to generate reports.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



NOTE: The input/output pin `SeqParxxGP_SC` is not applicable for `USERSEQ0GP` DFB.

Inputs

Input Parameter Description

Parameter	Type	Description
`InitCond-Fail`	`BOOL`	User can leverage on this input pin to connect to interlock conditions summary (`CONDSUMGP`) or other DFBs to take necessary actions depending on the value of `InitCondFail`.
`FailCond`	`BOOL`	User can leverage on this input pin to connect to detected failure conditions summary (`CONDSUMGP`) to take necessary actions depending on the value of `FailCond`.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
`SeqParxxGP_SC`	`SeqParxxGP_SC_DDT`, page 696	Provides the data needed to manage the parameters of a sequence from the control sequence itself or from a different external mechanism.
`SSCGP`	`SeqCTLGP_SC_DDT`, page 671	Provides the frequently needed data to monitor the sequence status and control it from the sequential control.
`SSTGP`	`SeqCTLGP_ST_DDT`, page 669	Provides the necessary data for monitoring and controlling sequence execution.
`SCFGGP`	`SeqCTLGP_CFG_DDT`, page 671	Provides the necessary data for background monitoring.
NOTE: The parameter `SeqParxxGP_SC` is not applicable for `USERSEQ0GP` DFB.		

Auxiliary Functions

What's in This Part

ASELECTGP - Analog Signal Selector 700
Message Box 712
Alarm Summary 722
Bool Setpoint 728
Real Setpoint 731
Integer Setpoint 734
Duration Setpoint 737
Scheduler Function 740

Overview

This part provides the functionality of the templates, a detailed description of the control (functions, pins, pin layout, and variables of the function blocks), and the supervision functions of auxiliary functions.

Auxiliary function blocks can be used in conjunction with other DFBs to provide additional services and data.

These function blocks do not reflect any specific installation.

⚠ WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate them.
- Review the implications of communication link interruptions and take actions to mitigate them.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, and applicable codes and regulations.
- Apply local accident prevention and safety regulations and guidelines.[1]
- Test each implementation of a system for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

ASELECTGP - Analog Signal Selector

What's in This Chapter

Template 700
Control 704
Supervision 707

Overview

This chapter describes the functionality of template, control services, and supervision functions of the Analog signal selector.

Template

Overview

This section describes the functionality, parameters, and interfaces of the $AnalogSelectGP template.

Description

General Description

The $AnalogSelectGP process control module template is used to select an analog signal among a maximum of four signals depending on the selection.

You can link this process template, for example, with the $PIDGP and $AnalogOutputGP, page 395 process templates by using the Links Editor.

Function Description

The main functions of the template are summarized in the following table:

Function	Description
Selection	The template selects the analog signal from among the input signals according to one of the following criteria: • Maximum • Minimum • Direct selection
Feedback	The template generates the signals needed to report to the templates. These templates generate the analog signals whether or not their signals have been selected. This functionality is especially useful when PID control is implemented in Override mode.
Downstream objects ($AnalogOutputGP/ $ControlValveGP/ $MotorVSGP) information to Advanced Controllers.	When Advanced Controllers (example, Advanced Process Control Library's Controllers and General Purpose Library's Advanced PID) are connected with Analog Actuators (example, ($AnalogOutputGP/$ControlValveGP/$MotorVSGP) via Analog Select block then the Downstream objects information are passed to Advanced controllers via ExtendedData interface of Analog Select template.

Parameters

Parameters

The AnalogSelectGP template defines operation of analog selector. Alternative analog selector input facets enable the modification of the analog selector that are available.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Setpoint	Force Remove Setpoint	Boolean	False	–
	Remote Setpoint Selection	Enum	Minimum (0)	0 is minimum of signals, 1 is SP1, 2 is SP2, 3 is SP3, 4 is SP4, 5 is maximum of signals

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Operation	Normal owner	Enum	Program (1)	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	Set owner	Enum	Operator (1)	Privilege to change owner. This parameter is used in Operator tab, page 709.
	Set setpoint	Enum	Operator (1)	Privilege to change setpoint. This parameter is used in Operator tab, page 709.
	Set reset		Operator (Confirmed) (10)	Privilege to reset. This parameter is used in Operator tab, page 709.
	Set parameters		Engineer (5)	Privilege to change parameters. This parameter is used in Operator tab, page 709.
	Set alarm acknowledge		Operator (1)	Privilege to set alarm acknowledge. This parameter is used in Alarm tab, page 67.
Trend	Sample period	Duraion	00:00:05	
	History length (weeks)	Integer	5	
	History rollover time	Duraion	12:00:00	
	History rollover day	String	Tuesday	
	Storage type		TRN_EVENT	
	Data location (optional)		[Data]:	

Asset and Display Parameters

For more details, refer to the topic describing the parameters of asset and display, page 43.

Data

The table describes the data parameters:

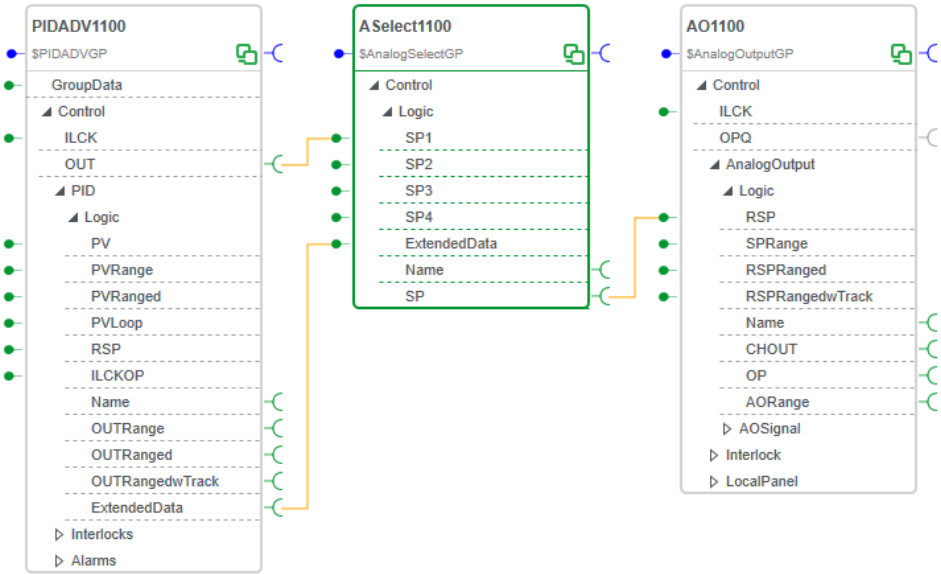
Section	Parameter description	Type	Default	Additional remarks
Deadband	Setpoint deadband	Integer	0	–
Description	Setpoint 1	String	@(Value1)	–
	Setpoint 2		@(Value2)	–
	Setpoint 3		@(Value3)	–
	Setpoint 4		@(Value4)	–
Range	Low range (Raw)	Float	0.0	–
	High range (Raw)	Float	100.0	–
	Low range (EU)	Float	0.0	–
	High range (EU)	Float	100	–
	Eng. units	String	%	–
	Format	String	####.#EU	–
Variable Tag Disable	Disable SP	Boolean	False	–
Alarm	Error description	String	@(Error)	–
	Error severity	Enum	Low(4000)	It allows to set the criticality (Critical (1000), High (2000), Medium (3000) and Low (4000)) of alarm.
	Error alarm group	String	Error	–
Trend Tag Disable	Disable SP trend	Boolean	False	–
Events	Enable log	Boolean	False	It allows to historize the change setpoints.
Historize	Historize status/ commands	Boolean	False	–
	Historize setpoint			–
	Historize setpoint 1			–
	Historize setpoint 2			–
	Historize setpoint 3			–
	Historize setpoint 4			–

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the $AnalogSelectGP (also applicable for $AnalogSelectCE) template as it appears in Links Editor and an example of other templates connected to it:



The following table lists the link interface that creates link symbol on faceplate:

Interface name	Runtime parameter
SP1	Setpoint1Navigate
SP2	Setpoint2Navigate
SP3	Setpoint3Navigate
SP4	Setpoint4Navigate
SP	SetpointNavigate

For more details, refer to Supervision chapter > Faceplate > Navigation Link of General Purpose Library User Guide.

Refer to common control module template interface, page 47 for more details on common interfaces.

The table describes the control module template interfaces:

Interface identifier	Type/Role	Description	Example
SP1	*$Real/Ref*	Setpoint 1 interface from other template.	A real variable connected to define different setpoint and output.
SP2	*$Real/Ref*	Setpoint 2 interface from other template.	
SP3	*$Real/Ref*	Setpoint 3 interface from other template.	
SP4	*$Real/Ref*	Setpoint 4 interface from other template.	
SP	*$Real/Def*	Current setpoint selected output interface to other template.	
ExtendedData	*ASelectINF/B*	ExtendedData interface is used to link Analog Select with Advanced Process Control Library's Controllers and General Purpose Library's Advanced PID.	ExtendedData interface is used to link Advanced PID and Analog Select

Control

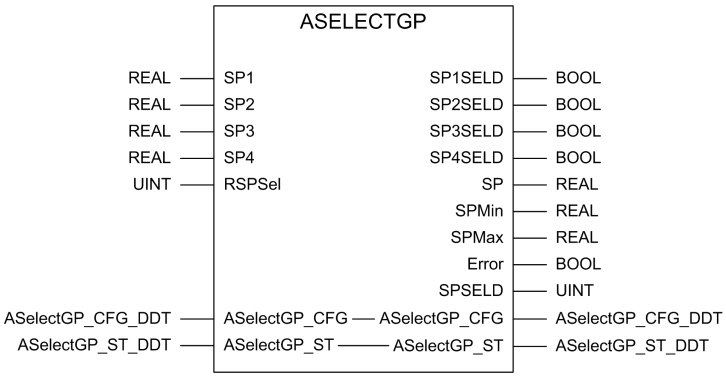
Overview

This section describes the ASELECTGP DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Name	Type	Description	
SP1..SP4	REAL	Value of the analog signals to be selected.	
RSPSel	UINT	Enables to select the selection method used:	
		RPSel	**SP** is calculated as:
		0	Minimum value among the enabled SPx signals (refer to the SPMin, page 705 output).
		1	SP1
		2	SP2
		3	SP3
		4	SP4
		5	Maximum value among the enabled SPx signals (refer to the SPMax, page 705 output).

Outputs

Output Parameter Description

Name	Type	Description
SP1SELD.. SP4SELD	BOOL	1 = Indicates the corresponding input (SPx) has been selected depending on the enabled signals and the selector value (refer to the RSPSel, page 704 input). No signal is activated if the selected signal (1..4) is not enabled or if the selector value is not valid (SPSel>5).
SP	REAL	Input value (SP1-SP4) selected with the SPSel input signal. Refer to the SP1SELD-SP4SELD outputs for more details. Set to 0 when none of these outputs is activated.
SPMin	REAL	Minimum input value (SP1-SP4) among those connected. Set to 0 if no output is connected.
SPMax	REAL	Maximum input value (SP1-SP4) among those connected. Set to 0 if no output is connected.
Error	BOOL	1 = Is activated if: • The RSPSel input value is out of range. • The selected signal is not connected. • No signal is selected. • No input signal is connected.
SPSELD	UINT	Selected selection method. Refer to the table for the RSPSel, page 704 input pin.

Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
ASelectGP_CFG	ASelectGP_CFG_DDT	Provides the data necessary to configure the DFB normally from the monitoring subsystem.
ASelectGP_ST	ASelectGP_ST_DDT	Provides the data needed to monitor and/or control the DFB status.

ASelectGP_CFG_DDT Type

Name	Type	Description
SP1	REAL	Refer to the SP1...SP4 input pins, page 704.
SP2	REAL	
SP3	REAL	
SP4	REAL	

ASelectGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	MinSELD	Bit0	BOOL	Minimum selected input (SPSELD = 0).
	SP1SELD	Bit1	BOOL	Refer to the SP1SELD, page 705 output pin.

Name	Bit Name	Bit	Type	Description
	SP2SELD	Bit2	BOOL	Refer to the SP2SELD, page 705 output pin.
	SP3SELD	Bit3	BOOL	Refer to the SP3SELD, page 705 output pin.
	SP4SELD	Bit4	BOOL	Refer to the SP4SELD, page 705 output pin.
	MaxSELD	Bit5	BOOL	Maximum selected input (SPSELD = 5).
	MinEn	Bit6	BOOL	Enables (1) the selection of the lowest SPx value. Enabled when more than one set-point is connected.
	SP1En	Bit7	BOOL	SP1 selection enabled. Enabled when a variable or value is connected to input SP1.
	SP2En	Bit8	BOOL	SP2 selection enabled. Enabled when a variable or value is connected to input SP2.
	SP3En	Bit9	BOOL	SP3 selection enabled. Enabled when a variable or value is connected to input SP3.
	SP4En	Bit10	BOOL	SP4 selection enabled. Enabled when a variable or value is connected to input SP4.
	MaxEn	Bit11	BOOL	Enables (1) the selection of the highest SPx value. Enabled when more than one set-point is connected.
	Error	Bit12	BOOL	Refer to the Error, page 705 output pin.
CFGW	–	–	WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
	Owner	Bit0	BOOL	Enables to configure whether the set-point selection parameters are set by the Program (0) or by the Operator (1).
	MinSel	Bit1	BOOL	Enables to select the minimum local set-point (Owner = 1) of the Operator from the monitoring subsystem. If the set-point is set by the Program (Owner = 0), the DFB continuously updates the value.
	SP1Sel	Bit2	BOOL	Enables to select the local SP1 set-point (Owner = 1) of the Operator from the monitoring subsystem. If the set-point is set by the Program (Owner = 0), the DFB continuously updates the value.
	SP2Sel	Bit3	BOOL	Enables to select the local SP2 set-point (Owner = 1) of the Operator from the monitoring subsystem. If the set-point is set by the Program (Owner = 0), the DFB continuously updates the value.
	SP3Sel	Bit4	BOOL	Enables to select the local SP3 set-point (Owner = 1) of the Operator from the monitoring subsystem. If the set-point is set by the Program (Owner = 0), the DFB continuously updates the value.
	SP4Sel	Bit5	BOOL	Enables to select the local SP4 set-point (Owner = 1) of the Operator from the monitoring subsystem. If the set-point is set by the Program (Owner = 0), the DFB continuously updates the value.
	MaxSel	Bit6	BOOL	Enables to select the maximum local set-point (Owner = 1) of the Operator from the monitoring subsystem. If the set-point is set by the Program (Owner = 0), the DFB continuously updates the value.
SP	–	–	REAL	Read-only access Refer to the output pin.

Public Variables

Public Variable Description

Variable	Type	Description
SC	ASelectGP_SC_DDT	Provides the frequently needed data to monitor and/or control the DFB status from the sequential control.

ASelectGP_SC_DDT Type

Name	Type	Description
Rem	BOOL	Read/write access 1 = Allows the DFB to be configured for a remote set-point -RSPSel. 0 = Allows the DFB to be configured for a local set-point -LSPSel- (0).
Owner	BOOL	Read-only access Refer to the ASelectGP_ST.CFGW.Owner input/output pin, page 705.
SP	REAL	Read-only access Refer to the SP output pin, page 705.
LSPSELD	UINT	Read-only access Enables the sequential control to assign the local set-point if the owner is the Program (Owner input/output is set to 0) and the selected set-point is the local one (SC.rem public variable is set to 0). Otherwise, the current set-point (SPSELD output) is continuously copied to this variable.
SP1SELD	BOOL	Read-only access Refer to the SP1SELD output pin, page 705.
SP2SELD	BOOL	Read-only access Refer to the SP2SELD output pin, page 705.
SP3SELD	BOOL	Read-only access Refer to the SP3SELD output pin, page 705.
SP4SELD	BOOL	Read-only access Refer to the SP4SELD output pin, page 705.
Error	BOOL	Read-only access Refer to the Error output pin, page 705.

Supervision

Overview

This section describes the Supervision functions and runtime services that are available for the ASELECTGP.

Supervision Functions

Description

Core and optional resources provide the following monitoring and operation functions:

- Core functions:
  - Owner selection
  - Selector operation

Genies

Genie Properties

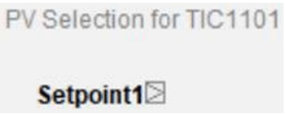
Refer to Genie Properties, page 62

Genie Icons

Refer to Genie Icons, page 63

Representation

The table shows the genies of the `gpl_analogselect` library:

Graphic Symbol	Genie Name
	Aselectdiscription
	Aselectindicator

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

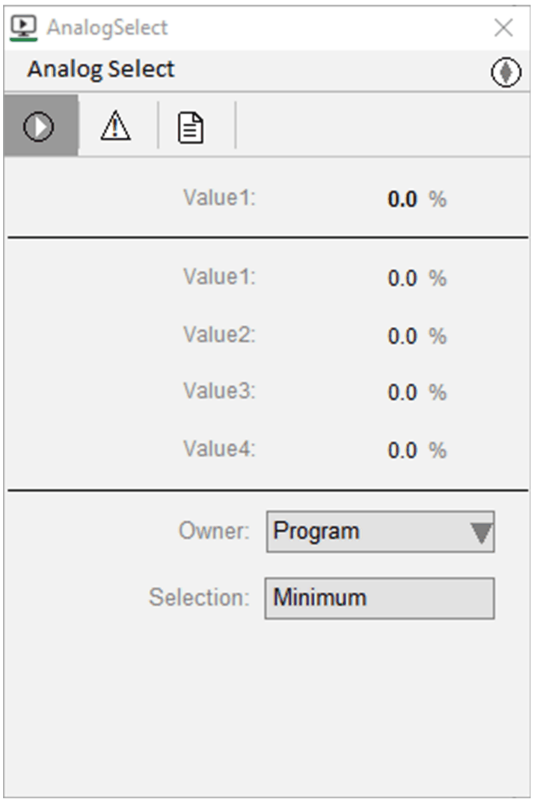
Available Tabs

During operation, clicking an `ASELECTGP` genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Alarm, page 67

Operator Tab

The figure shows an operator tab:



The table describes the control function of an operator tab:

Label	Item Name	Description	Security Parameter
Current Owner	OwnerSelect	Current owner of ASELECTGP. Refer to Owner, page 58.	–
Selection	OperatorSet-point	Allows you to change the setpoint, if owner = operator.	Priv_Mode

Items

Overview

This section describes the variables, equipstatus and equipstate variable, abnormal and action required conditions, alarms, events, and trend item of ASELECTGP.

Variables

The table describes the variable items that are used by Supervision components:

Item Name	Description	Enumeration	Address
Bit items derived from StatusWord.			
SelectedMinimum	Minimum setpoint selection status	Selected/Not Selected	*Bit0*
SelectedInput1	Setpoint 1 selection status		*Bit1*
SelectedInput2	Setpoint 2 selection status		*Bit2*
SelectedInput3	Setpoint 3 selection status		*Bit3*

Item Name	Description	Enumeration	Address
SelectedInput4	Setpoint 4 selection status		*Bit4*
SelectedMaximum	Maximum setpoint selection status		*Bit5*
EnableMinimum	Minimum setpoint enabled	Enabled/Disabled	*Bit6*
EnableInput1	Setpoint 1 connection status		*Bit7*
EnableInput2	Setpoint 2 connection status		*Bit8*
EnableInput3	Setpoint 3 connection status		*Bit9*
EnableInput4	Setpoint 4 connection status		*Bit10*
EnableMaximum	Maximum setpoint enabled		*Bit11*
Error	Error status	Active/Inactive	*Bit12*
Bit items derived from ConfigurationWord.			
OwnerSelect	Owner of control module	Operator/Program	*Bit0*
SelectMinimum	Minimum setpoint selection	Selected/Not Selected	*Bit1*
SelectInput1	Setpoint 1 selection		*Bit2*
SelectInput2	Setpoint 2 selection		*Bit3*
SelectInput3	Setpoint 3 selection		*Bit4*
SelectInput4	Setpoint 4 selection		*Bit5*
SelectMaximum	Maximum setpoint selection		*Bit6*
IO Items			
StatusWord	Status word	–	ASELECT1_ST.STW
ConfigurationWord	Configuration word		ASELECT1_ST.CFGW
Calculated Items			
OperatorSetpoint	Operator setpoint	–	GPL_ AnalogSelectOperator-Setpoint
Abnormal	Shows abnormal condition of the ASELECTGP Refer to Abnormal Conditions, page 710.	–	GPL_RatioCtlAbnormal

Abnormal Conditions

The table describes the abnormal conditions of ASELECTGP:

Item Name	Description	Address
Bit items derived from Abnormal variable word.		
AbnormalOwner	Abnormal owner status appears if the current owner is not a normal owner which configured at the instance level.	*Bit0*
AbnormalOperatorTab	Abnormal status of operator tab. This condition appears if the default owner is changed.	*Bit8*

Action Required Conditions

The table describes the action required conditions of ASELECTGP:

Item Name	Description	Address
Bit items derived from ActionRequired variable word.		
ActionOperatorTab	Action required status of operator tab.	*Bit8*

Events

Events are conditions which are detected during the operation of control module or the logging of operator actions:

Item Name	Description
Status	All changes in the ASELECTGP status and output.
Mode	All changes in the mode and owner.
Action	All actions by the operator.

Trends

No trend item is used by Supervision components for ASELECTGP.

Message Box

What's in This Chapter

Template 712
Control 714
Supervision 719

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the message box.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$MessageBoxGP` template.

Description

General Description

The `$MessageBoxGP` process control module template offers a standard user interface mechanism and allows messages for the operator to be displayed. It allows requesting confirmations and/or data without having to configure customized interfaces to channel the messages.

The template, when combined with other templates designed for this purpose, allows you to configure the format of the message from the program in the Control Participant. The format can include icons, buttons, and data entry fields.

Function Description

The main functions of the template are described in the following table:

Function	Description
Message Display Management	The templage allows the message display to be activated/aborted. Only one message is managed at a time. Therefore, the number of `$MessageBoxGP` templates instantiated needs to be equal to the number of simultaneous messages required per processing unit.
Icons	Allows you to select the icon that needs to be shown on the user interface. Enables to display the following icons: • Stop: To show diagnostics messages, for example. • Question mark: To show request messages for data entry, for example. • Exclamation mark: To show alert messages, for example. • Information: To show steps, or the status of the process in general, for example.

Function	Description
Buttons	Enables you to select the buttons that apply: • No buttons required (for alert messages, for example). • One single **Ok** button (for example, for manual action confirmation messages). • An **Ok** button and a **Cancel** button (for example, for messages in which the user needs to choose between two alternatives).
Optional Data	You can show up to 2 pieces of data stored in the controller along with the message. A request for the entry of those pieces of data can also be issued; for example, for entering process sampling results.

Parameters

Parameters

The `$MessageBoxGP` template defines operation of messagebox. Alternative messagebox facets enable the modification of the messagebox that are available.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Deadband	Deadband of complementary data 1	`Integer`	0	–
	Deadband of complementary data 2	`Integer`	0	–
Range	High range value of complementary data 1	`Float`	99999.99	–
	Low range value of complementary data 1		-99999.99	–
	Engineering units of complementary data 1	`String`	–	–
	Display format of complementary data 1		#####.##	–
	Low range value of complementary data 2	`Float`	-99999.99	–
	High range value of complementary data 2		99999.99	–
	Engineering units of complementary data 2	`String`	–	–
	Display format of complementary data 2		#####.##	–
Security	Set operate	`Enum`	Operator (1)	Privilege to change operate. This parameter is used in Operator Tab, page 720.
Variable Tag Disable	Disable data 1	`Boolean`	False	–
	Disable message for data 1			–
	Disable data 2			–
	Disable message for data 2			–
Historize	Historize status/ commands	`Boolean`	False	It allows to historize the change in status or commands.

Section	Parameter description	Type	Default	Additional remarks
	Historize data			It allows to historize the change in data.

Asset and Display Parameters

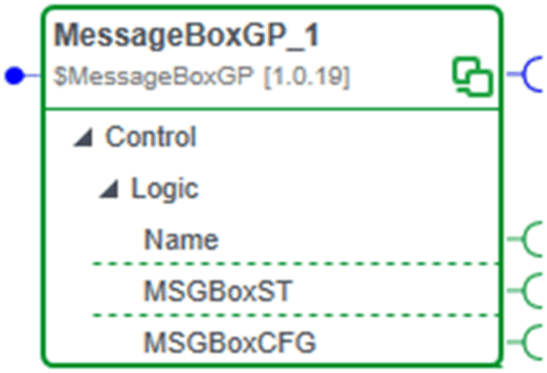
For more details, refer to the topic describing the parameters of asset and display, page 43.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$MessageBoxGP` (also applicable `$MessageBoxCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



For interface identifiers, refer to the control module template interfaces, page 49.

Control

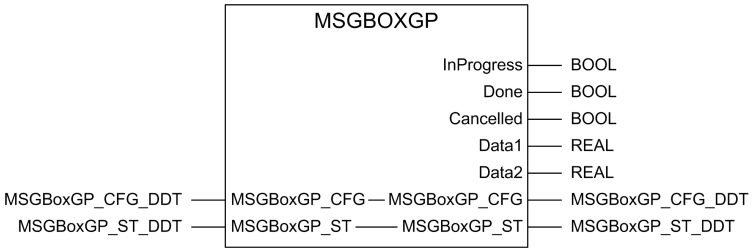
Overview

This chapter describes the `MSGBOXGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

The function module does not have inputs. Its operation parameters are configured through the SC public variable and the MSGBoxGP_CFG and MSGBoxGP_ST data exchange structures.

Outputs

Output Parameter Description

Name	Type	Description
InProgress	BOOL	1 = This signal is kept activated while a message is waiting for response, that is, if a request for a new message has been issued (SC.Show = 1 and SC.CODE <> 0) and none of the selected buttons have been clicked on yet (**Ok** –MSGBoxGP_ST.CFGW.8- or **Cancel** – MSGBoxGP_ST.CFGW.9-) or if an abort request has been issued (SC.Abort) for the message (the only option to finalize the message for alert messages without waiting for an operator action).
Done	BOOL	1 = This signal is activated upon accepting the message in progress, that is, when the **Ok** button is clicked on MSGBoxGP_ST.CFGW.Ok input/output pin, page 715 while a message is pending for response (InProgress =1). The signal is deactivated (0) when the message in progress is aborted or a new message is in progress.
Cancelled	BOOL	1 = This signal is activated upon canceling the message in progress, that is, when the **Cancel** button is clicked on MSGBoxGP_ST.CFGW.CANCEL input/output pin, page 715 while a message is pending for response (InProgress = 1). The signal is deactivated (0) when the message in progress is aborted or if a new message is in progress.
Data1	REAL	Enables to check the first optional data shown to the operator and/or entered by the operator according to the type of message selected. This data is not exchanged until it has been confirmed by the operator (any values that could have possibly been entered previously during editing are not copied). This signal is only applicable if MSGBoxGP_ST.STW.Data1En is equal to 1.
Data2	REAL	Enables to check the second optional data shown to the operator and/or entered by the operator according to the type of message selected. This data is not exchanged until it has been confirmed by the operator (any values that could have possibly been entered previously during editing are not copied). This signal is only applicable if MSGBoxGP_ST.STW.Data2En is equal to 1.

Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
MSGBoxGP_CFG	MSGBoxGP_CFG_DDT	Provides the additional data (Data 1 and/or Data 2) that can be associated with the message in progress.
MSGBoxGP_ST	MSGBoxGP_ST_DDT	Provides the data necessary for displaying the message, carrying out the applicable actions (**Ok** and/or **Cancel**) and defining the message to be shown.

MSGBoxGP_CFG_DDT Type

The following table describes the MSGBoxGP_CFG_DDT type:

Name	Type	Description
Data1	REAL	Read/write access Data 1 associated with the message. Loaded with SC.Data1 when the latter is applicable (refer to the SC.Mode public variable, page 717) when showing the message. And vice-versa, the value in this signal is copied to SC.Data1 after the message has been confirmed, provided that the data is applicable (refer to the SC.Mode public variable, page 717).
Data2	REAL	Read/write access Data 2 associated with the message. Loaded with SC.Data2 when the latter is applicable (refer to the SC.Mode public variable, page 717) when showing the message. And, vice-versa, the value in this signal is copied to SC.Data2 once the message has been confirmed, provided that the data is applicable (refer to the SC.Mode public variable, page 717).
MSG1	STRING [40]	Read/write access Data 1 supplementary message. Limited to 39 characters. Loaded with SC.MSG1 when the latter is applicable (refer to the SC.Mode public variable, page 717) when showing the message.
MSG2	STRING [40]	Read/write access Data 2 supplementary message. Limited to 39 characters. Loaded with SC.MSG2 when the latter is applicable (refer to the SC.Modepublic variable, page 717) when showing the message.

MSGBoxGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description			
STW	–	–	WORD	Provides the data necessary to display the message. Reproduces the values entered in SC.Mode (type of message to be shown).			
	OkEn	Bit0	BOOL	Indicates that the **Ok** button is applicable to the message in progress.			
	OkCancelEn	Bit1	BOOL	Indicates that the **Ok** and **Cancel** buttons are applicable to the message in progress.			
	Data1En	Bit2	BOOL	Indicates that Data 1 is applicable to the message in progress.			
	Data2En	Bit3	BOOL	Indicates that Data 2 is applicable to the message in progress.			
	Icon1En	Bit4	BOOL	Refer to the MSGBoxGP_ST.STW.Icon1En input/output.			
	Icon2En	Bit5	BOOL	Refer to the MSGBoxGP_ST.STW.Icon2En input/output.			
	Icon3En	Bit6	BOOL	Combined with MSGBoxGP_ST.STW.Icon1En and MSGBoxGP_ST.STW.Icon2En, enables to select the icon to be shown with the message. Refer to the SC.Mode public variable, page 717.			
				Icon1En	**Icon2En**	**Icon3En**	**Icon**
				OFF	OFF	OFF	No icons are displayed
				OFF	OFF	ON	Information icon
				OFF	ON	OFF	Question icon
				OFF	ON	ON	Exclamation icon
				ON	OFF	-	Stop icon
				ON	ON	-	Exclamation icon

Name	Bit Name	Bit	Type	Description
CFGW	–	–	WORD	Enables the applicable actions (**Ok** and/or **Cancel**) to be carried out.
	Ok	Bit8	BOOL	Setting this signal to 1 enables to **Ok** the message. The DFB sets this signal to 0 after the signal has been processed.
	Cancel	Bit9	BOOL	Setting this signal to 1 enables to **Cancel** the message. The DFB sets this signal to 0 after the signal has been processed.
MSG	–	–	STRING [40]	Read-only access Main message to be shown. The character string is limited to 39 characters.

Public Variables

Public Variable Description

Variable	Type	Description
SC	MSGBoxGP_SC_DDT	Provides the interface for configuring the messages that are to be shown, as well as their format (buttons, icons, and/or additional data).

The following table describes the MSGBoxGP_SC_DDT type:

Name	Type	Description
MSG	STRING [40]	Read/write access Defines the message to be shown. The character string is limited to 39 characters. The value set in this character string is transferred to MSGBoxGP_ST.MSG the moment that message activation (SC.Show = 1) is requested.
MSG1	STRING [40]	Read/write access Defines supplementary message 1 to be shown. It is applicable as a supplementary message for the main message, or as a descriptive message for data 1. The character string is limited to 39 characters. The value defined in this character string is transferred to MSGBoxGP_CFG.MSG1 the moment that message activation (SC.Show = 1) is requested.
MSG2	STRING [40]	Read/write access Sets the desired supplementary message 2 to be shown. It is applicable as a supplementary message for the main message, or as a descriptive message for data 2. The character string is limited to 39 characters. The value defined in this character string is transferred to MSGBoxGP_CFG.MSG2 the moment that message activation (SC.Show = 1) is requested.
Mode	UINT	Read/write access Refer to the Mode, page 718 signal.
Ok	BOOL	Refer to the Ok Inputs/Outputs, page 715 pin.
OkCancel	BOOL	1 = Enables the ok and cancel buttons.
AddData1	BOOL	Additional data 1.
AddData2	BOOL	Additional data 2.
Icon1	BOOL	1 = Stop icon.
Icon2	BOOL	1 = Question icon.
Icon3	BOOL	1 = Exclamation icon if stop and question icons are true.
Icon4	BOOL	1 = Information icon.
Data1	REAL	Read/write access Needs to be set with the initial value to be shown in data 1 of the message if this data is applicable when the message is activated (SC.Show = 1). Also, it allows the value entered by the user to be

Name	Type	Description
		captured after the message is confirmed with the **Ok** button (MSGBoxGP_ST.CFGW.Ok).
Data2	REAL	Read/write access Needs to be set with the initial value to be shown in data 2 of the message if this data is applicable when the message is activated (SC.Show = 1). Also, it allows the value entered by the user to be captured after the message is confirmed with the **Ok** button (MSGBoxGP_ST.CFGW.Ok).
Show	BOOL	Read/write access 1 = Enables the message to be activated. 0 = The DFB sets the signal to 0 after the signal has been processed. Prior to activation of this signal, the text of the message to be shown (SC.MSG) and its format (SC.Mode) needs to be defined.
Abort	BOOL	Read/write access 1 = Enables the display of the message in progress to be aborted. 1 = The DFB sets the signal to 0 after the latter is processed. It is used both to stop showing display-only messages and to abort messages that are awaiting the operator action.
InProgress	BOOL	Read-only access Refer to the InProgress output pin, page 715.
Done	BOOL	Read-only access Refer to the Done output pin, page 715.
Cancelled	BOOL	Read-only access Refer to the Cancelled output pin, page 715.

Mode Signal

Enables to configure the type of message that needs to be displayed and configure the icons (Stop, Question, Exclamation, or information), buttons (**Ok** and/or **Cancel**), and/or additional data (up to 2) that needs to be shown together with the message. The configuration loaded in this signal is transferred to the low byte under the MSGBoxGP_ST.STW status word the moment that message activation (SC.Show = 1) is requested.

The Mode signal is defined by adding the codes associated with the different format types to those that can be shown in the message.

The codes as well as an example of their use is included in the following table:

Code	Format
1 (2^0)	Enable the **Ok** button
2 (2^1)	Enable the **Ok** and **Cancel** buttons
4 (2^2)	Additional data 1 is applicable
8 (2^3)	Additional data 2 is applicable
16(2^4)	Show the Stop icon
32 (2^5)	Show the Question icon
48 (2^4 and 2^5)	Show the Exclamation icon
64 (2^6)	Show the Information icon

For example, to configure the with: **Ok** button,

(* Data 1, Data 2, and Question icon *)

```
MSG1_MSGBoxGP.SC.Mode:= 1 + 4 + 8 + 32;
```

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the `MSGBOX`.

Genies

Genie Properties

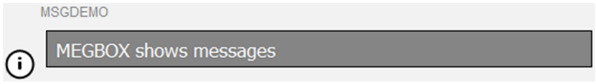
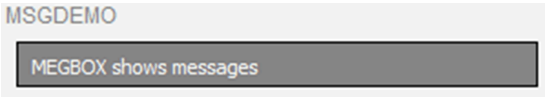
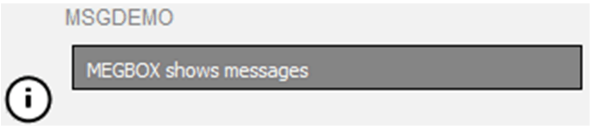
Refer to Genie Properties, page 62.

Genie Icons

Refer to Genie Icons, page 63.

Representation

The table describes the genies of the `gpl_msgbox` library:

Graphic Symbol	Genie name	Description
	*Msgboxlargetexticon*	Message with icons in large text.
	*Msgboxnormaltext*	Message without icons in normal text.
	*Msgboxnormaltexticon*	Message with icons in normal text.

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

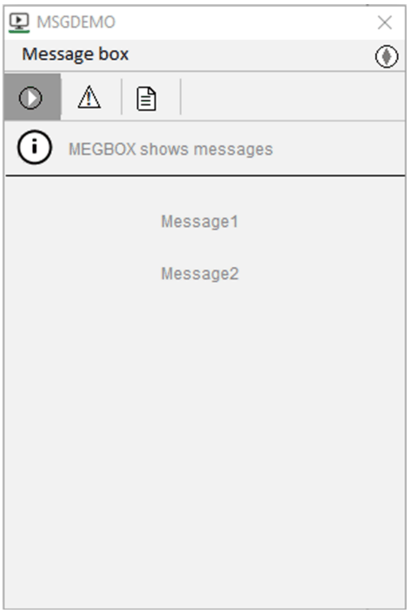
Available Tabs

During operation, clicking on message box genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Alarm, page 67

Operator Tab

The figure shows an operator tab:



Items

Overview

This section describes the variables and equipstatus variables of `message box`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`OKEnabled`	OK button enable status to the current message	Enabled/Disabled	*Bit0*
`OKCancelEnabled`	OK and cancel button enable status for the current message		*Bit1*
`Data1Enabled`	Data 1 enable status for the current message		*Bit2*
`Data2Enabled`	Data 2 enable status for the current message		*Bit3*
`Icon1Enabled`	Icon 1 enable status for the current message		*Bit4*
`Icon2Enabled`	Icon 2 enable status for the current message		*Bit5*
`Icon3Enabled`	Icon 3 enable status for the current message		*Bit6*
Bit items derived from ConfigurationWord.			
`OK`	OK command	Selected/Not Selected	*Bit8*
`Cancel`	Cancel command		*Bit9*

Item name	Description	Enumeration	Address
IO Items			
StatusWord	Status word	–	MSGBOXGP_ST.STW
ConfigurationWord	Configuration word		MSGBOXGP_ST.CFGW
Calculated Items			
IconStatus	Icon status of the message box	–	GPL_ MessageBoxIconSta- tus

Alarm Summary

What's in This Chapter

Template 722
Control 724
Supervision 724

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the alarm summary.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$AlarmSummaryGP` template.

Description

General Description

The `$AlarmSummaryGP` process control module template is the combination of the `$CONDSUMGP` and the `$DigitalInputGP` process templates.

It allows you to enable and disable the evaluation of alarms with up to 15 external alarm conditions.

Function Description

The main functions of the template are described in the following table:

Function	Description
Alarm	You can enable or disable the evaluation of an alarm and indicate the status that needs to be considered for the alarm. You need to incorporate an external logic to evaluate this alarm.

Parameters

Parameters

The `$AlarmSummaryGP` template defines operation of alarm summary. Alternative alarm summary facets enable the modification of the alarm summary that are available.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	PV state to be considered as bad	Boolean	TRUE	–
	watch alarm	Enum	Refinement(1)	–
	PV state to be considered as alarm		True Value (3)	–
Time	set point for on delay	Duration	00:00:00	–
	set point for off delay			–

Alarm

For more details, refer to the topic describing the parameters of condsum, page 74.

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Security	Set mode - override	Enum	Supervisor (2)	Privilege to change override mode. This parameter is used in Engineering Tab, page 726.
	Set mode - maintenance		Maintenance (3)	Privilege to set maintenance mode. This parameter is used in Engineering Tab, page 726.
	Set reset		Operator (Confirmed) (10)	Privilege to reset. This parameter is used in Operator Tab, page 726.
	Set alarm acknowledge		Operator (1)	Privilege to acknowledge alarm. This parameter is used in Alarm Tab, page 67.
	Set bypass conditions		Supervisor (2)	Privilege to set bypass conditions mode. This parameter is used in Engineering Tab, page 726.

Data

The table describes the data parameters:

Section	Parameter description	Type	Default	Additional remarks
Alarm	Alarm failure description/channel failure alarm description	String	@(Alarm Failure)/@ (Channel Failure Alarm)	–
	Alarm failure alarm severity/channel failure alarm severity	Enum	Low(4000)	It allows you to set the criticality (Critical (1000), High (2000), Medium (3000) and Low (4000)) of alarm.
	Alarm failure alarm group/channel failure alarm group	String	Failure	–
Events	Enable log	Boolean	False	It allows to historize the change setpoints.
Historize	Historize status/ commands	Boolean	False	–

Section	Parameter description	Type	Default	Additional remarks
	Historize conditions			–

Asset and Display Parameters

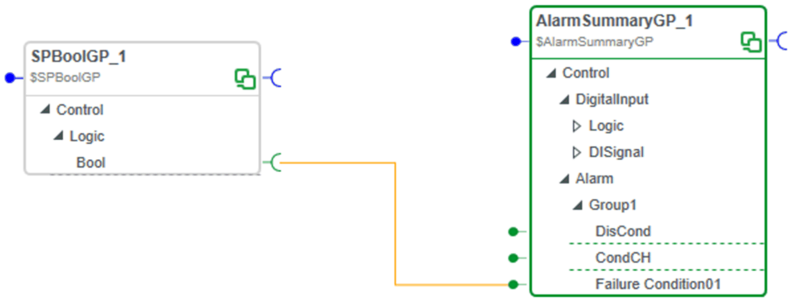
For more details, refer to the topic describing the parameters of asset and display, page 43.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$AlarmSummaryGP` (also applicable for `$AlarmSummaryCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



For interface identifiers, refer to the control module template interfaces, page 49.

Control

Overview

The alarm summary when implemented with `DInputGP` control provide data to the supervision components.

Description

For details about alarm summary control, refer to `Digital Input, page 416` chapter.

Supervision

Overview

This section describes the Supervision functions and runtime services that are available for the `AlarmSummary`.

Supervision Functions

Description

Core and optional resources provide the following monitoring and operation functions:

- Core functions:
  - Alarm configuration
  - Enabling/disabling of alarms
  - Simulation mode
  - Management of individual alarm conditions

Genies

Genie Properties

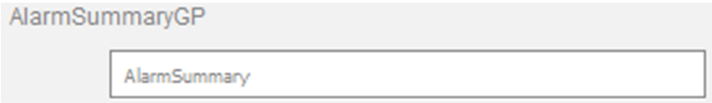
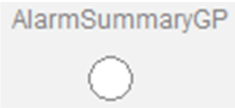
Refer to Genie Properties, page 62

Genie Icons

Refer to Genie Icons, page 63

Representation

The table shows the genies of the `gpl_alarmsummary` library:

Graphic Symbol	Genie Name
	Alarmtext
	Circle

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

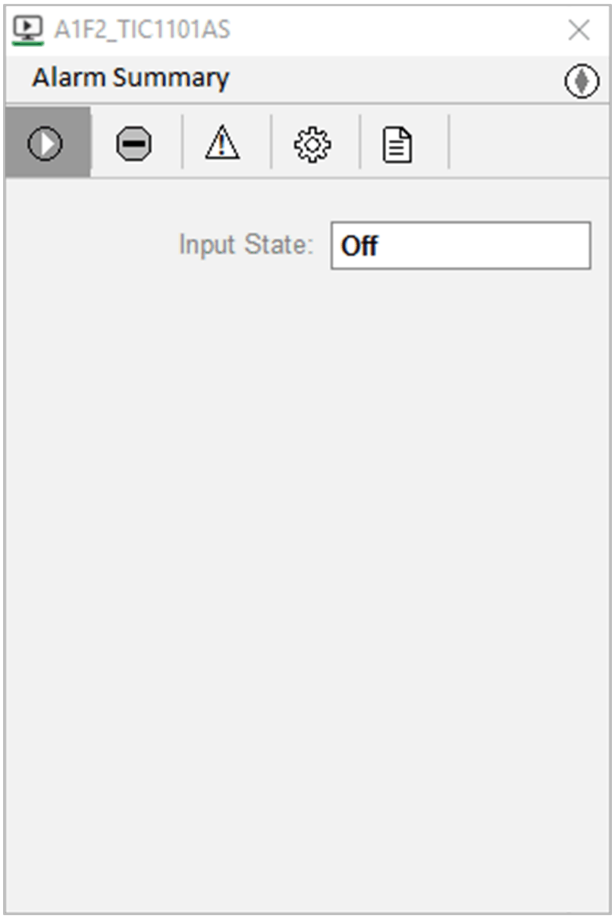
Available Tabs

During operation, clicking an `AlarmSummary` genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Engineering
  - Alarm, page 67
  - Event, page 69
- Tabs for optional functions, which appear only if configured:
  - Failure, page 78

Operator Tab

The figure shows an operator tab:

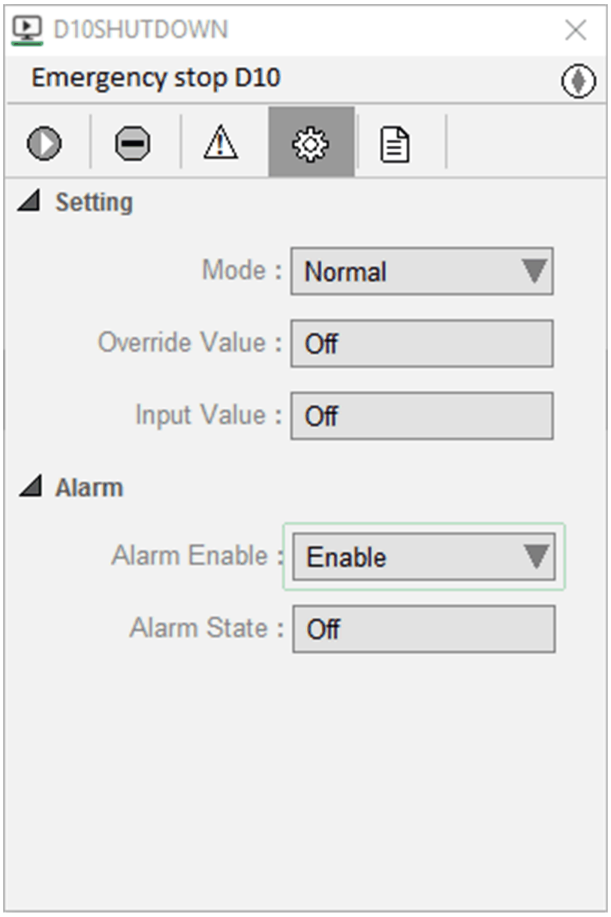


The table describes the control function of an operator tab:

Label	Item Name	Description	Security Parameter
Input State	State	Input status of alarm summary.	–

Engineering Tab

The figure shows an engineering tab:



The table describes the control function of engineering tab:

Label	Item name	Description	Security parameter, page 723
Setting			
Mode	Mode	Allows you to view and change the mode. Refer to operating mode, page 59.	Set mode - override, set mode - maintenance. **NOTE:** Each configurable mode has the different security parameters
Override Value	OverrideValue	Value will be considered when input source is operator.	Set mode - override
Input Value	ChannelInputValue	Channel value	–
Alarm			
Alarm Enable	AlarmEnable	Alarm monitoring	–
Alarm State	State	Process value	

Items

Overview

The table describes the variable items that are managed by Supervision components of `AlarmSummary`.

Name	Description
`DINPUTGP`	Refer to `DINPUTGP`, page 422.
`CONDSUMGP`	Refer to `CONDSUMGP`, page 78.

Bool Setpoint

What's in This Chapter

Template 728
Supervision 729

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the `$SPBool`.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$SPBoolGP` template.

Description

General Description

The `$SPBoolGP` process control module template is used to enter a setpoint/value from the Supervision runtime.

The template creates a variable of data type `BOOL` whose value you can set by using various types of genies that are referenced by the template.

The output varies depending on the genie that you select.

Parameters

Parameters

The `$SPBoolGP` template defines operation of Setpoint `BOOL`.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Initial Value of Variable in form of Bool	`Boolean`	FALSE	–

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Selection for the Genie	`Enum`	Pulse (0)	It allows you to set the configuration Pulse (0), Latch (1), Checkbox (2).
	Time for the pulse to active	`Integer`	5	Value in seconds.
Security	Set operate	`Enum`	Operator (1)	Privilege to change operator.

Asset and Display Parameters

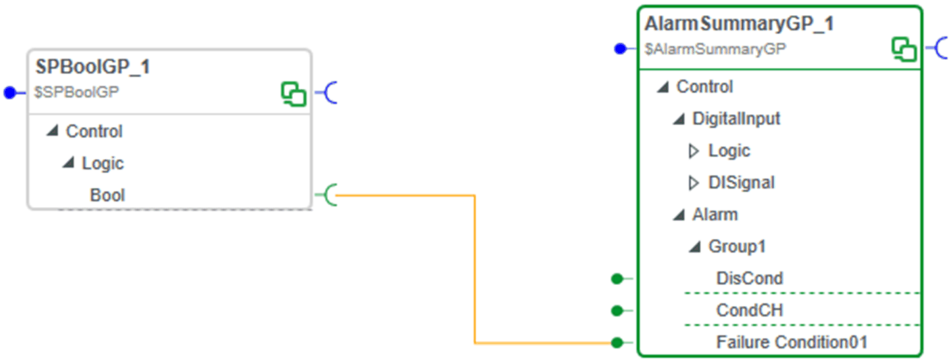
For more details, refer to the topic describing the parameters of asset and display, page 43.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This is an example of an instance of `$SPBoolGP` (also applicable for `$SPBoolCE`) template as it appears in `Links Editor`.



For interface identifiers, refer to the control module template interfaces, page 49.

Supervision

Overview

This section describes the Supervision functions and runtime services that are available for the `SPBool`.

Genies

Genie Properties

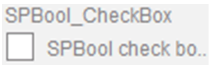
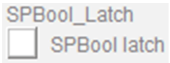
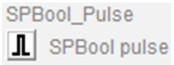
Refer to Genie Properties, page 62

Genie Icons

Refer to Genie Icons, page 63

Representation

The table shows the genies of the `gpl_spmanage` library:

Graphic symbol	Genie name
	Spbool_checkbox
	Spbool_latch
	Spbool_pulse

Items

Overview

No items are available for `SPBool`.

Real Setpoint

What's in This Chapter

Template 731
Supervision..... 732

Overview

This chapter describes the Template, Supervision functions and runtime services that are available for the `$SPREALGP`.

Template

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the `$SPRealGP` template.

Description

General Description

The `$SPRealGP` process control module template is used to enter a setpoint/value from the Supervision runtime.

The template creates a variable of data type `REAL` whose value you can set by using various types of genies that are referenced by the template.

An entry of up to 10 digits is allowed for the value. You can use up to 2 decimal places.

Parameters

Parameters

The `$SPRealGP` template defines operation of real data.

Contorl

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Initial value of variable in real format	`Float`	0.0	–

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Raw zero range	`String`	0.0	–
	Raw full range		100.0	–
	Eng. unit		%	–
	Format		###.#	–
Security	Set Operate	`Enum`	Operator (1)	Privilege to change operator.

Asset and Display Parameters

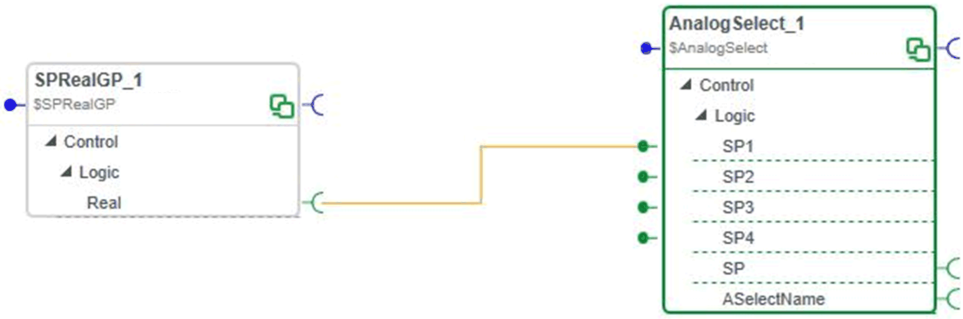
For more details, refer to the topic describing the parameters of asset and display, page 43.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$SPRealGP` (also applicable for `$SPRealCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



The table describes the control module template interfaces:

Interface identifier	Type/Role	Description	Example
`Real`	*$Real/Def*	Real interface.	

Supervision

Overview

This section describes the Supervision functions and runtime services that are available for the `SPReal`.

Genies

Genie Properties

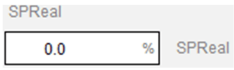
Refer to Genie Properties, page 62

Genie Icons

Refer to Genie Icons, page 63

Representation

The table shows the genies of the `gpl_spmanage` library:

Graphic Symbol	Genie Name
	Realsetpoint

Items

Overview

No items are available for `SPReal`.

Integer Setpoint

What's in This Chapter

Template 734
Supervision 735

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the `$SPINTGP`.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$SPIntGP` template.

Description

General Description

The `$SPIntGP` process control module template is used to enter a setpoint/value from the Supervision runtime.

The template creates a variable of data type `INTEGER` whose value you can set by using various types of genies that are referenced by the template.

NOTE: The corresponding EcoStruxure Process Expert data type that is used for data entry is SHORT .

Parameters

Parameters

The `$SPIntGP` template defines operation of integer data.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Initial value of variable in integer format	`Short`	0	–

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Raw zero range	`Integer`	0	–
	Raw full range		100	–
	Eng. unit	`String`	%	–
	Format		#####EU	–
Security	Set operate	`Enum`	Operator (1)	Privilege to change operator.

Asset and Display Parameters

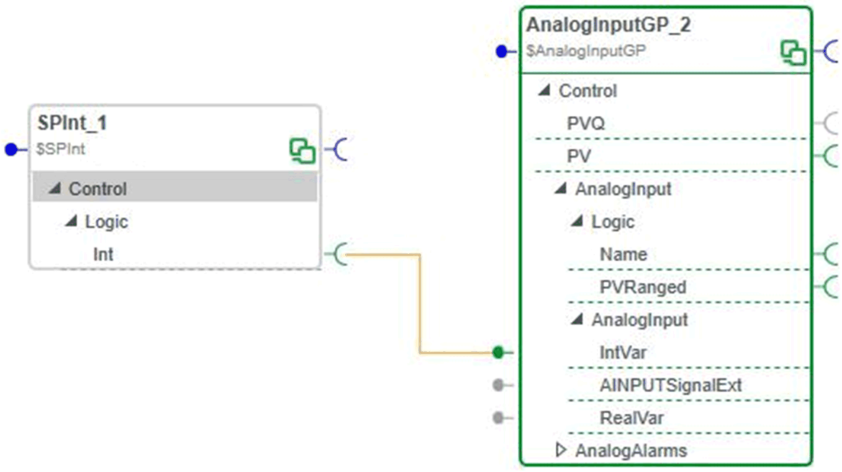
For more details, refer to the topic describing the parameters of asset and display, page 43.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$SPIntGP` (also applicable for `$SPIntCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



For interface identifiers, refer to the control module template interfaces, page 49.

Supervision

Overview

This section describes the Supervision functions and runtime services that are available for the `SPInteger`.

Genies

Genie Properties

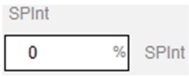
Refer to Genie Properties, page 62

Genie Icons

Refer to Genie Icons, page 63

Representation

The table shows the genies of the `gpl_spmanage` library:

Graphic Symbol	Genie Name
	Integersetpoint

Items

Overview

No items are available for `SPInteger`.

Duration Setpoint

What's in This Chapter

Template 737
Supervision 738

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the `$SPDurationGP`.

Template

Overview

This section describes functionality of the `$SPDurationGP` template.

Description

General Description

The `$SPDurationGP` process control module template is used to enter a setpoint/value from the Supervision runtime.

The template creates a variable of data type `TIME` whose value you can set by using various types of genies that are referenced by the template.

NOTE: The corresponding EcoStruxure Process Expert data type that is used for data entry is Duration.

The equivalent data type of the Supervision Participant is ULONG.

Parameters

Parameters

The `$SPDurationGP` template is used to enter a setpoint/value from the supervision runtime.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Initial Value of Variable in Time format	`Duration`	00:00:00	–

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Security	Set Operate	`Enum`	Operator (1)	Privilege to change operator.

Asset and Display Parameters

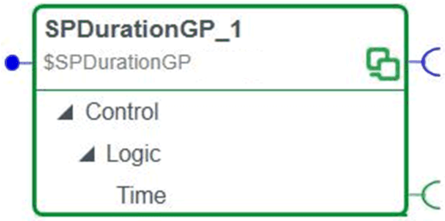
For more details, refer to the topic describing the parameters of asset and display, page 43.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$SPDurationGP` (also applicable for `$SPDurationCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



For interface identifiers, refer to the control module template interfaces, page 49.

Supervision

Overview

This section describes the Supervision functions and runtime services that are available for the `SPDuration`.

Genies

Genie Properties

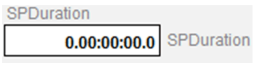
Refer to Genie Properties, page 62

Genie Icons

Refer to Genie Icons, page 63

Representation

The table shows the genies of the `gpl_spmanage` library:

Graphic Symbol	Genie Name
	spduration

Items

Overview

This section describes the variables of `SPDuration`.

Variables

The following variable tag parameters define the range of the setpoint values that you can enter and need to comply with the *ULONG* data type:

- Eng Zero Scale: Low end of the range for the setpoint.
- Eng Full Scale: High end of the range for the setpoint.

These parameter values need to be compatible with the *ULONG* data type that is used by the tag.

Scheduler Function

What's in This Chapter

Template 740
Control 742
Supervision 745

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the `Scheduler`.

Template

This section describes the functionality, parameters, and interfaces of the `$SchedulerGP` template.

Description

General Description

The `$SchedulerGP` template is used to schedules based on events or time driven schedules. The template allows user to select up to 10 different schedules which can be configured either event or time schedule

Functional Description

The main functions of motor template are described in the following table:

Function	Description
Time scheduling	The template compares the current PLC time with configured event times in the Schedule data. After comparison, the function block releases the outputs.
Event scheduling	The template compares the status of the event as per the event configured in template and if the event is active, then function block releases the outputs.
Scheduler status	Active or Inactive The scheduler module is in operation when scheduler status is set to active on faceplate.
Mode selection	The object can be switched to different modes of operation like On, Off, and Pulse.
Pulse timing	If user selects the mode as Pulse, then user can select the Pulse On time and Off time either in seconds, minutes, or hours based on the selection in Pulse Time In on the faceplate.

Parameters

The `$SchedulerGP` template provides different control and supervision parameters to the user to control the functions as per the requirement.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Number of schedule group	Enum	1(2)	–
	Schedule {x} type		Time (0)	–
NOTE: {x} represents from 0 to 10.				

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Security	Set setpoint	–	Operator (1)	Privilege to change setpoint. This parameter is used in Operator Tab., page 747.
	Set parameters	–	Engineer (5)	Privilege to change setpoint. This parameter is used in Operator Tab., page 747.
Data				
Configuration	Format of set value	String	#####.#EU	–
	Schedule {x} description		@(Time Schedule {x})	–
NOTE: {x} represents from 0 to 10.				

Asset and Display Parameters

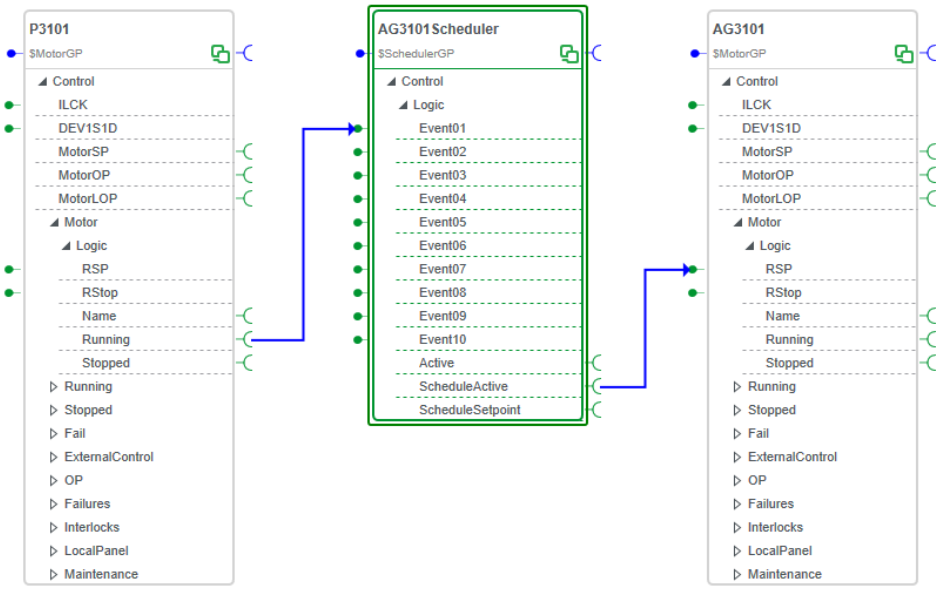
For more details, refer to the topic describing the parameters of asset and display, page 43.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$SchedulerGP` template as it appears in `Links Editor` and an example of other templates connected to it:



For interface identifiers, refer to the control module template interfaces, page 49.

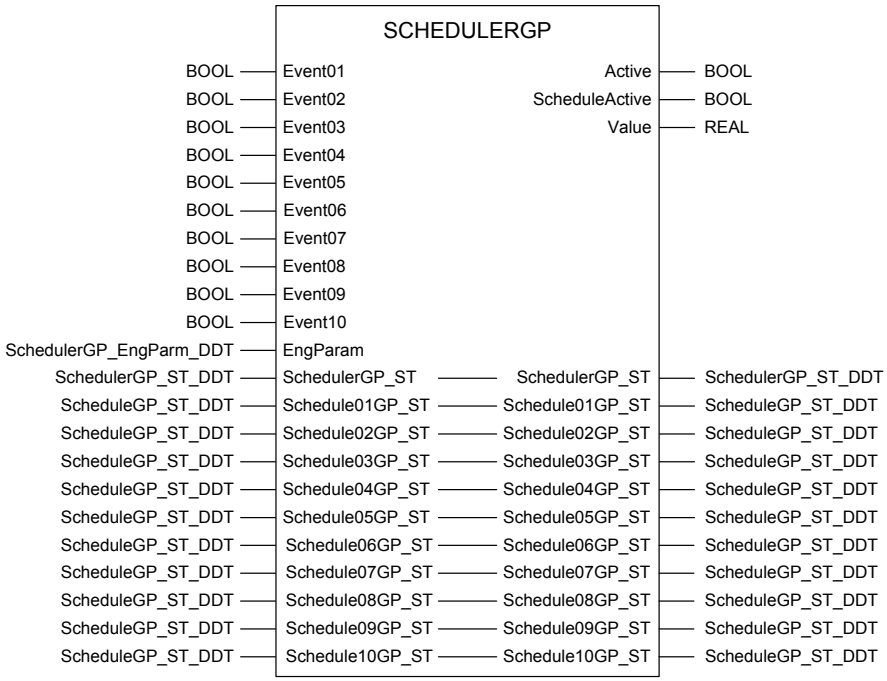
Control

This chapter describes the `SCHEDULERGP` DFB.

DFB Representation

Overview

This figure shows the FBD representation of the `SCHEDULERGP` DFB.



Inputs

Input Parameter Description

Name	Type	Description
Event {x}	BOOL	1 = Indicate to DFB that Event condition {y} is active. **NOTE:** If the respective EngParam. Schedule {x} Mode = 1, then Event condition consider to activate schedule.
EngParam	SchedulerGP_EngParm_DDT	Engineering parameters
NOTE: {x} represents from 01 to 10 and {y} represents from 1 to 10.		

SchedulerGP_EngParm_DDT

The following table describes the structure of the derived data type **`SchedulerGP_EngParm_DDT`** for the `SCHEDULERGP` DFB:

Name	Type	Description
Schedule {X} Mode	BOOL	Indicated the mode of schedule {y} mode. 0 = Time-Based Schedule 1= Event-Based Schedule
NOTE: {x} represents from 01 to 10 and {y} represents from 1 to 10.		

Outputs

Output Parameter Description

Parameter	Type	Description
Active	BOOL	1 = Indicates the scheduler active status.
ScheduleAc-tive	BOOL	Indicates state as per scheduled output. 1 = Schedule is On 0 = Schedule is Off
Value	REAL	Real value output changes as per the predefined scheduled value when active = 1.

Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
SchedulerGP_ST	SchedulerGP_ST_DDT	Common data structure of DFB status and supervision.
Schedule{X}GP_ST	ScheduleGP_ST_DDT	Schedule {Y} data structure.
NOTE: {X} represents from 01 to 10 and {Y} represents from 1 to 10.		

SchedulerGP_ST_DDT

The following table describes the structure of the derived data type SchedulerGP_ST_DDT for the SCHEDULERGP DFB:

Name	Bit Name	Bit	Type	Description
CurrentSchedule	–	–	INT	Current active schedule number.
CurrentSetValue	–	–	REAL	Current active schedule set value.
EventStatus	–	–	WORD	Scheduler event status.
	SchedulerAc-tive	Bit0	BOOL	Scheduler status. 0 = Inactive 1= Active
	Event{X} Active	Bit {Y}	BOOL	Refer to Input Pin, page 743
ScheduleModeSta-tus	–	Bit0	WORD	Schedule mode selection status.
	Schedule{X} ModeSel	Bit {Y}	BOOL	Refer to SchedulerGP_EngParm_DDT, page 743
NOTE: {X} represents from 01 to 10 and {Y} represents from 1 to 10.				

Schedule{X}GP_ST_DDT

The following table describes the structure of the derived data type ScheduleGP_ST_DDT for the SCHEDULERGP DFB:

Name	Bit Name	Bit	Type	Description
PresetValue	–	–	REAL	Sets the value for schedule.
StartTime	–	–	REAL	Sets the start time of schedule.
OnTime	–	–	DINT	Sets the schedule on time for mode 2...4
OffTime	–	–	DINT	Sets the event pause time for mode 2...4
Day	–	–	INT	Sets the day on which schedule has to be performed.
Mode	–	–	INT	Sets the type of schedule: 0 = Off 1 = On 2 = Pulse in seconds 3 = Pulse in minutes 4 = Pulse in hours **NOTE:** SchedulerGP_EngParam_DDT (Schedule {X} Mode = 1 then applicable modes are): 1 = On 2 = Pulse in seconds 3 = Pulse in minutes 4 = Pulse in hours
NOTE: {X} represents from 01 to 10.				

Supervision

This section describes the Supervision functions and runtime services that are available for the SCHEDULERGP.

Genies

Genie Properties

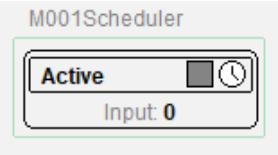
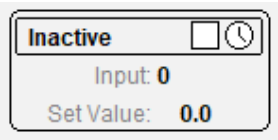
Refer to Genie Properties, page 62

Genie Icons

Refer to Genie Icons, page 63

Representation

The table shows the genies of the gpl_scheduler library:

Graphic Symbol	Genie name	Description
	Scheduler	Active scheduler with input value
	Scheduler_SV	Inactive scheduler with input value

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a general description of the graphic elements that are used in the faceplate.

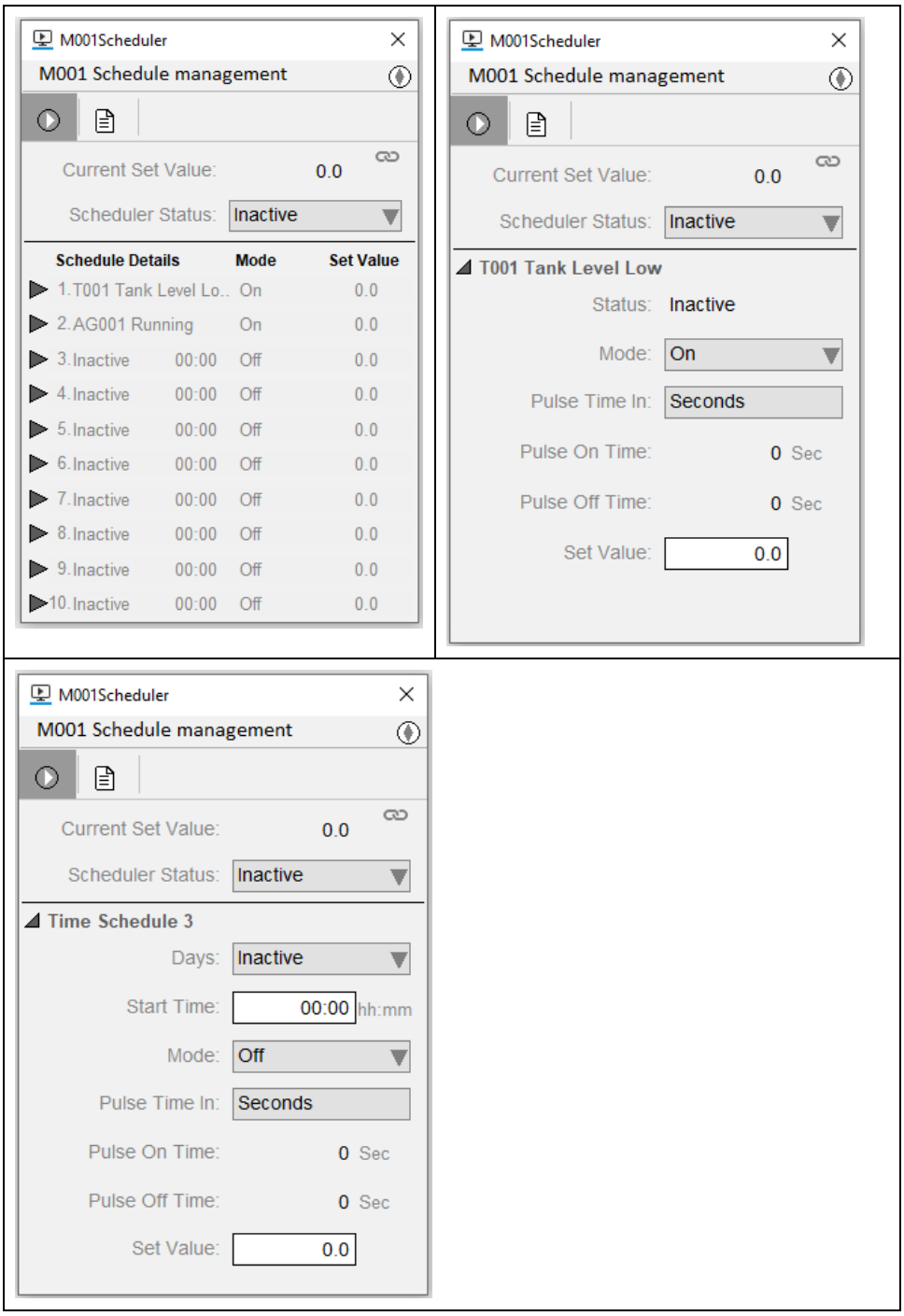
Available Tabs

During operation, clicking an `Scheduler` genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Event, page 69

Operator Tab

The figure shows an operator tab:



The table describes the control function of an operator tab:

Label	Item name	Description	Security parameter, page 741
Current Set Value	CurrentSetVal-ue	Current set value	–
Scheduler Status	SchedulerAc-tive	Scheduler status	–
T001 Tank Level Low			
Status	Event1Active	Event status	–
Mode	Schedule1-Mode	Schedule mode condition	–

Label	Item name	Description	Security parameter, page 741
Pulse Time In	PulseEngUnit1	Schedule 1 pulse time engineering unit	–
Pulse On Time	OnTime1	Schedule 1 pulse On n time	–
Pulse Off time	OffTime1	Schedule 1 pulse Off time	–
Set value	SetValue1	Set value	–
Time Schedule 3			
Days	Day3	Schedule 3 day setting	–
Start Time	StartTime3	Schedule 3 start time	–

Items

Overview

This section describes the EventStatus and ResponseStatus variables, and IO items of `Scheduler`.

Variables

The table describes the variable items that are used by Supervision components:

Item Name	Description	Enumeration	Address
Bit items derived from EventStatus			
`SchedulerActive`	Scheduler command	Active/In	*Bit0*
`Event{x}Active`	Event {x} status		*Bit{x}*
NOTE: {x} represents from 1 to 10.			
Bit items derived from ResponseStatus.			
`Schedule1Mode`	Schedule 1 mode condition	Operator/Program	*Bit{y}*
NOTE: {x} represents from 1 to 10 and {y} represents from 0 to 9.			
IO Items			
`CurrentSchedule`	Current schedule	–	`Scheduler_SchedulerGP_ST.CurrentSchedule`
`CurrentSetValue`	Current set value		`Scheduler_SchedulerGP_ST.CurrentSetValue`
`SetValue{x}`	Schedule {x} set value	–	`Scheduler_Schedule05GP_ST.PresetValue`
`StartTime{x}`	Schedule {x} start time		`Scheduler_Schedule01GP_ST.StartTime`
`OnTime{x}`	Schedule {x} pulse On time	–	`Scheduler_Schedule01GP_ST.OnTime`
`OffTime{x}`	Schedule {x} pulse Off time		`Scheduler_Schedule01GP_ST.OffTime`
`Day{x}`	Schedule {x} day setting	–	`Scheduler_Schedule01GP_ST.Day`
`Mode{x}1`	Schedule {x} mode setting		`Scheduler_Schedule01GP_ST.Mode`

Item Name	Description	Enumeration	Address
PulseMode{x}	Schedule {x} pulse time mode setting	–	Scheduler_Schedule01GP_ST_PulseMode
PulseEngUnit{x}	Schedule {x} pulse time engineering unit		Scheduler_Schedule01GP_ST_PulseEngUnit1
NOTE: {x} represents from 1 to 10.			

Events

Events are conditions which are detected during the operation of control module or the logging of operator actions:

Item Name	Description
Status	All changes in the ASELECTGP status and output.
Mode	All changes in the mode and owner.
Action	All actions by the operator.

Calculations

What's in This Part

Digital Calculation 751
Analog Calculation 755
Analog Linear 761

Overview

This part the provides the functionality of the templates, a detailed description of the control (functions, pins, pin layout, and variables of the function blocks), and the supervision functions of `ACALCGP`, `ALINEARGP`, and `DCALCGP` templates.

These function blocks do not reflect any specific installation.

⚠ WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate them.
- Review the implications of communication link interruptions and take actions to mitigate them.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, and applicable codes and regulations.
- Apply local accident prevention and safety regulations and guidelines.[1]
- Test each implementation of a system for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

Digital Calculation

What's in This Chapter

Template 751
Control 752

Overview

This chapter describes the functionality of template, control services, and supervision functions of the `DCALCGP` template.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$DigitalCalcGP` template.

Description

General Description

The `$DigitalCalcGP` process control module template is used to perform calculations based on a digital signal. The calculations for frequency and totalizing are performed.

The template stores up to twenty samples with rising edges in the measurement (present value (`PV`)) based on which the calculations are performed.

Functional Description

The main functions of the template are described in the following table:

Function	Description
Frequency	The frequency at which the input signal oscillates is calculated.
Totalizing	The totalization is calculated by counting pulses and scaling to engineering units.

NOTE: Template uses the FREERUN function for time calculations. Therefore, the `$DCalcGP` template is only available if the controller CPU supports this function.

Parameters

Parameters

The `$DigitalCalcGP` template defines operation of digital calculation.

Control

The table describes the control parameters:

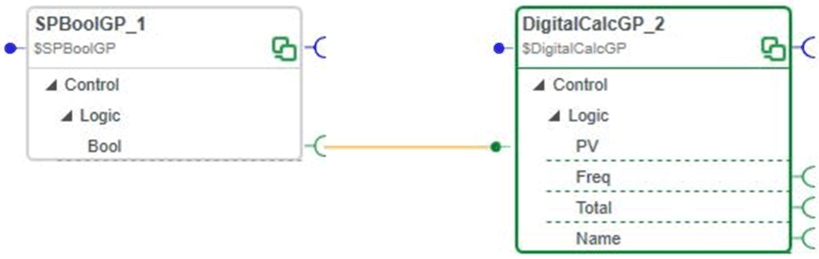
Section	Parameter description	Type	Default	Additional remarks
Configuration	Maximum Number of Samples (between 2 to 20)	`Unsigned-Short`	20	Maximum number of samples that need to be stored and that are used in the calculations.
	Frequency Conversion Coefficient Applied to the Frequency Calculation	`Float`	1.0	Conversion factor applied to the frequency calculation. Allows the frequency to be calculated in the desired engineering units (for example, RPM).
	Totalized Conversion Coefficient Applied to the Totalizing Calculation			Conversion factor applied to the totalizing calculation output. Allows the total to be calculated in the desired engineering units.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$DigitalCalcGP` (also applicable for `$DigitalCalcCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



For interface identifiers, refer to the control module template interfaces, page 49.

Control

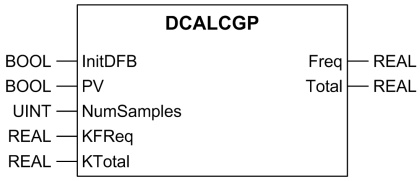
Overview

This section describes the `DCALCGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
InitDFB	BOOL	0 = Rising edge on this input is detected, the DFB internal timers are reset and the first sample (time and value) is stored in the memory. In the same execution cycle, the value of the outputs is generated. **NOTE:** The input is not reset (0) by the DFB.
PV	BOOL	1 = Value of the measurement. **NOTE:** You need to verify that the execution characteristics of the DFB are compatible with the input frequency of the signal; so, no edges are lost and the accuracy of the frequency calculation is adequate for the requirements of the process.
NumSamples	UINT	Maximum number of samples that need to be stored and that are used in the calculations. Configuration range: From 2 to 20 Default value: 20 **NOTE:** During the first runs of the DFB, a smaller number of samples than those configured may be used.
KFReq	REAL	Conversion factor applied to the frequency calculation (refer to the Freq, page 753 output). Allows the frequency to be calculated in the desired engineering units (for example, RPM). If this signal is 0, the DFB applies a default factor of 1.0.
KTotal	REAL	Conversion factor applied to the totalizing calculation Total, page 753 output. Allows the total to be calculated in the desired engineering units. If this signal is 0, the DFB applies a default factor of 1.0.

Outputs

Output Parameter Description

Parameter	Type	Description
Freq	REAL	The frequency is calculated (in Hz when KFreq is 1.0) based on the time samples in which the rising edges of the measurement have taken place (PV). Frequency calculation is based on the following formula.
Total	REAL	Calculates the totalization while it is activated (refer to the SC.TotalEn public variable, page 754). A unit is accumulated (scaled with KTotal, when applicable) each time a rising edge occurs on the input signal. The calculation is only made if it has been enabled and you can set it to 0 (refer to SC public variable, page 754).

Frequency calculation formula:

$$KFREQ \times \left(\frac{\sum_{i=1}^{NUMSAMPLES} \frac{i^2}{t_i}}{\sum_{i=1}^{NUMSAMPLES} i} \right)$$

Where:

KFreq	Conversion factor applied to the frequency calculation. Refer to the `KFReq` input, page 753.
n	Number of samples (up to 20 after the whole internal sample table is loaded).
t	Relative time elapsed from the first edge to the corresponding edge.

Checking the Saturation Limit of the Output

Calculations based on values of type REAL may give incorrect results. For example, if the increment (*KTotal*) that is used in the calculation of the output is smaller than the precision of the output, the output may be wrong.

You can check whether the output is correct by using the *SATURATION* function block.

For more information, refer to *SATURATION* in the help of Control Participant.

Public Variables

Public Variable Description

Variable	Type	Description
`SC`	`DCalcGP_SC_DDT`	Provides the frequently needed data to monitor and control the DFB.

`DCalcGP_SC_DDT` Type

Name	Type	Description
`PV`	`BOOL`	Read-only access Refer to the `PV` input pin, page 753.
`Freq`	`REAL`	Read-only access Refer to the `Freq` output pin, page 753.
`Total`	`REAL`	Read-only access Refer to the `Total` output pin, page 753.
`TotalEn`	`BOOL`	Read/write access 1 = Enables the totalizing calculation. 0 = Disables the totalizing calculation. Does not set the totalizer to 0.
`TotalRst`	`BOOL`	Read/write access 1 = Resets the totalized value (`Total` output). 0 = After processing, the DFB sets this signal to 0.

Analog Calculation

What's in This Chapter

Template 755
Control 757

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the `ACALCGP`.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$AnalogCalcGP` template.

Description

General Description

The `$AnalogCalcGP` process control module template is used to perform calculations based on a analog signal.

The process template can perform the following calculations:

- Maximum
- Minimum
- Average
- Totalizing (integral with the trapezoidal rule)
- Derivative rate of change (ROC)

The template stores up to twenty samples with rising edges in the measurement (present value (`PV`)) based on which the calculations are performed.

Function Description

The main functions of the template are described in the following table:

Function	Description
Derivative	The slope of the measurement development is calculated with linear regression.
Totalizing	The totalization (for example, for flows) is calculated with the trapezoidal rule algorithm.
Average	Calculates the average of the values memorized in the template.
Maximum	Calculates the maximum value of the measurement among those stored in the template.
Minimum	Calculates the minimum value of the measurement among those stored in the template.The template allows alarm monitoring to be enabled/disabled individually.

NOTE: The template block uses the Freerun function for time calculations. Therefore, the `ACALCGP` template is only available if the controller CPU supports this function.

Parameters

Parameters

The `$AnalogCalcGP` template defines operation of a typical analog calculation. Alternative analog calculation facets enable the modification of the analog calculation that are available.

Control

The table describes the control parameters:

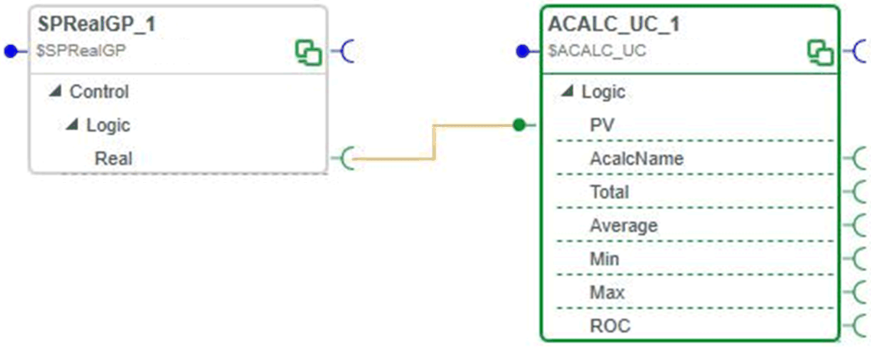
Section	Parameter description	Type	Default	Additional remarks
Configuration	Scan Rate	`Duration`	00:00:0-1	Allows you to configure the sample acquisition time. The template takes a sample each time a time equal to or greater than the one configured for this input has elapsed. For calculation purposes, the time elapsed between one template execution and another is taken into account.
	Maximum Number of Samples (between 2 to 20)	`Unsigned-Short`	20	Maximum number of samples that need to be stored and that are used in the calculations.
	Coefficient for Gradient Applied to the Derivative Calculation	`Float`	1.0	Conversion factor applied to the derivative calculation output. Allows the frequency to be calculated in the desired engineering units.
	Totalized Conversion Coefficient Applied to the Totalizing Calculation			Conversion factor applied to the totalizing calculation output.Allows the total to be calculated in the desired engineering units.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$AnalogCalcGP` (also applicable for `$AnalogCalcCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



For interface identifiers, refer to the control module template interfaces, page 49.

Control

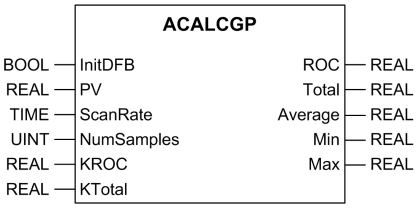
Overview

This section describes the `ACALCGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
`InitDFB`	`BOOL`	0 = Indicates rising edge on this input is detected, the DFB internal timers are reset and the first sample (time and value) is stored in the memory. **NOTE:** The input is not reset (0) by the DFB.
`PV`	`REAL`	Calculated value of the measurement in engineering units.
`ScanRate`	`TIME`	Allows you to configure the sample acquisition time. The DFB takes a sample each time a time equal to or greater than the one configured for this input has elapsed. For calculation purposes, the

Parameter	Type	Description
		time elapsed between one DFB execution and another is taken into account. **NOTE:** You need to verify that the DFB execution parameters are compatible with the time configured in the `ScanRate` parameter; this allows the DFB to run with a high enough frequency to carry out sampling with the configured time including latency times. In any case, the DFB considers any possible execution delays regarding the theoretical time adjusted in this parameter when it comes to the calculations.
`NumSamples`	`UINT`	Maximum number of samples that need to be stored and that are used in the calculations. Configuration range: From 2 to 20 Default value: 20 **NOTE:** During the first runs of the DFB, a smaller number of samples than those configured may be used.
`KROC`	`REAL`	Conversion factor applied to the derivative calculation `ROC` output pin, page 758. Allows the derivative to be calculated in the desired engineering units. If this signal is 0, the DFB applies a default factor of 1.0.
`KTotal`	`REAL`	Conversion factor applied to the totalizing calculation `Total` output pin, page 758. Allows the totalization to be calculated in the desired engineering units. If this signal is 0, the DFB applies a default factor of 1.0.

Outputs

Output Parameter Description

Parameter	Type	Description
`ROC`	`REAL`	The derivative is calculated (in engineering units/sec for a `KROC` of 1.0) through linear regression. Calculation is based on the formula provided in this topic.
`Total`	`REAL`	Totalization is calculated with a trapezoidal sum between the current and the previous sample (Therefore, it is only calculated starting with the second sample and is 0 in the meantime) Calculation is based on the formula provided in this topic.
`Average`	`REAL`	Calculates the average value of the stored samples. It is calculated as the measurement is present at the `PV` input during the first run.
`Min`	`REAL`	Calculates the minimum value among the stored samples. It is calculated as the measurement is present at the `PV` input during the first run.
`Max`	`REAL`	Calculates the maximum value among the stored samples. It is calculated as the measurement is present at the `PV` input during the first run.

Calculation formula for the derivative:

$$KROC \times \left(\frac{n\sum(t \times PV) - \sum t \times \sum PV}{n\sum t^2 - \left(\sum t\right)^2} \right)$$

Where:

KROC	Conversion factor applied to the derivative calculation. Refer to the KTotal input, page 757.
n	Number of samples (up to 20 after the whole internal sample table is loaded).
t	Relative time elapsed from the first sample to the sample used in the calculation (corresponding PV value).
PV	Sample value.

Calculation formula for the totalization:

$$KTOTAL \times (t_1 - t_0) \times \left(\frac{PV_1 + PV_0}{2} \right)$$

Where:

KTotal	Refer to the KROC input, page 757.
t_1	Time in which the current sample is taken.
t_0	Time in which the previous sample was taken.
PV_1	Value of the current measurement.
PV_0	Value of the previous measurement.

The calculation is carried out with a value relative to 0, that is, values under 0 will be deducted from the cumulative calculation. The calculation is only performed if it is enabled and resetting is available (refer to the SC public variable, page 759).

Public Variables

Public Variable Description

Variable	Type	Description
SC	ACALCGP_SC_DDT	Provides the data frequently needed data to monitor the status of the alarms from the sequential control.

ACALCGP_SC_DDT Type

Name	Type	Description
PV	REAL	Read-only access Refer to the PV input pin, page 757.
ROC	REAL	Read-only access Refer to the ROC output pin, page 758.
Total	REAL	Read-only access Refer to the Total output pin, page 758.
Average	REAL	Read-only access Refer to the Average output pin, page 758.
Min	REAL	Read-only access Refer to the Min output pin, page 758.
Max	REAL	Read-only access Refer to the Max output pin, page 758.

Name	Type	Description
`TotalEn`	`BOOL`	Read/write access 1 = Enables the totalizing calculation 0 = Disables the totalizing calculation. Does not set the totalizer to 0.
`TotalRST`	`BOOL`	Read/write access 1 = Resets the totalized value (`Total` output pin). 0 = The signal is processed.

Analog Linear

What's in This Chapter

Template 761
Control 762

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the `ALINEARGP`.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$AnalogLinearGP` template.

Description

General Description

The `$AnalogLinearGP` process control module template is used to transform a signal by using a linear characterization function. The characterization function is defined with the use of points (up to five).

Parameters

Parameters

The `$AnalogLinearGP` template defines operation of analog linear. Alternative analog linear facets enable the modification of the analog linear that are available.

Control

The table describes the control parameters:

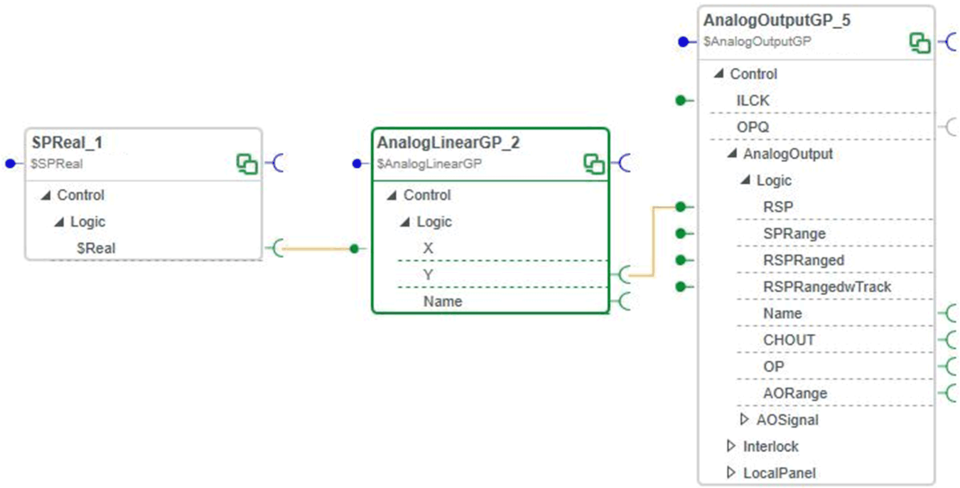
Section	Parameter description	Type	Default	Additional remarks
Configuration	X value defines the characteristics of function for point (1...5)	`Float`	0.0	–
	Y value defines the characteristics of function for point (1...5)	`Float`	0.0	–
	Number of points used (2...5)	`Unsigned-Short`	2	Maximum number of samples that need to be stored and that are used in the calculations.

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$AnalogLinearGP` (also applicable for `$AnalogLinearCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



For interface identifiers, refer to the control module template interfaces, page 49.

Control

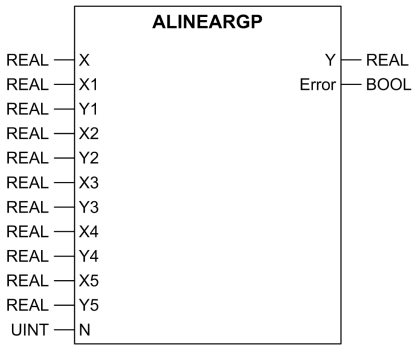
Overview

This section describes the `ALINEARGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
X	REAL	Value of the analog signal to be transformed.
X1	REAL	X value in point 1.
Y1	REAL	Y value in point 1.
X2	REAL	X value in point 2.
Y2	REAL	Y value in point 2.
X3	REAL	X value in point 3.
Y3	REAL	Y value in point 3.
X4	REAL	X value in point 4.
Y4	REAL	Y value in point 4.
X5	REAL	X value in point 5.
Y5	REAL	Y value in point 5.
N	UINT	Number of points used (2...5).

Outputs

Output Parameter Description

<table>
<tr><th>Parameter</th><th>Type</th><th>Description</th></tr>
<tr><td>Y</td><td>REAL</td><td>Value calculated from the X signal and the characterization function defined with the points (X1, Y1) to (X5, Y5).

It is calculated as followed:
<table>
<tr><td>Y=Y1</td><td>for <= X1</td></tr>
<tr><td>Y=f(X)</td><td>by linear interpolation according to the points entered for X >X1 and X< Xn</td></tr>
<tr><td>Y=Yn</td><td>for X>=Xn

Where n corresponds to the number of points selected on the N input. If the N input is less than or equal to 2, Y is unconditionally assigned to Y2. If N is greater than 5, the characterization function is applied as if the value were 5.</td></tr>
</table></td></tr>
<tr><td>Error</td><td>BOOL</td><td>1 = High-level alarm.

The parameter is set to 1 if:
• The N input is not within range.
• The (X,Y) points are not correctly ordered (X axis).</td></tr>
</table>

Equipment Module

What's in This Part

Pump Set Equipment Module 765
Pump Set 785
Auxiliary Functions of Pump Set Equipment Module 804
Flow Control Equipment Module 807
Flow Control 817
Auxiliary Functions of Flow Control Equipment Module 836
Equipment Module 839
`EMCTLGP` - Equipment Module Controller 852
`EMCTLGP` - Example of Use 871
EMLOGICGP - User Defined Process Logic 881
Auxiliary Functions of Equipment Module 883

Overview

This part provides the functionality of the templates, a detailed description of the control (functions, pins, pin layout, and variables of the function blocks), and the Supervision functions of equipment module.

These components do not reflect any specific installation.

⚠ WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate them.
- Review the implications of communication link interruptions and take actions to mitigate them.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, and applicable codes and regulations.
- Apply local accident prevention and safety regulations and guidelines.[1]
- Test each implementation of a system for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

Pump Set Equipment Module

What's in This Chapter

Description of Pump Set Equipment Module 765
Strategy Selection and Functionality 766
Disabling the Strategies Based on Equipment Module State 767
Input and Output Parameters 768
Input Parameters Applicable For Each Strategy 770
Pump Set Equipment Module State Machine 770
Pump Set Equipment Module Diagnosis State 771
Pump Set Pumping Asset Diagnosis State 772
Modifying the PumpSetPattern Template 772

Overview

This chapter describes the Pump Set Equipment Module management.

Description of Pump Set Equipment Module

Overview

It is an unprotected function block which is derived from `EMLOGIC` user-defined DFB. and `PUMPINGASSET05GP` DFB The `PUMPSETPATTERN` DFB checks the status and controls the operation of control modules (Motors/Valves/Sensors, etc.) and services (Initial Conditions/Detected Failure Conditions, etc.) to form a Pump Set equipment module.

A pump, an inlet valve, an outlet valve, and a drain valve are collectively referred to as a pumping asset. To configure a pumping asset, connecting a pump is mandatory whereas the valves are optional.

The `PUMPSETPATTERN` DFB along with `PUMPINGASSET05GP` DFB can be used to run common pumping applications like Sump management, Transfer and Circulation pump, etc. The maximum number of pumping assets supported for operation is 10. The number of pumping assets can be modified according to the users application requirement.

Function Description

The main functions of the DFB are described in the following table:

Function	Description
Incorrect Configuration	Determines the total number of pumps connected to the PumpingAsset05GP from the connections to the pump input pins of the DFB. If the pump connections are not in order, it detects an incorrect configuration. For example, If `PUMP01`, `PUMP02`, `PUMP04` input pins are connected, the incorrect configuration is detected since the pumps are not connected sequentially. When there is a incorrect configuration is detected, `PUMPSETPATTERN` equipment is not available for operation.
Order Sequence	The order sequence arranges the pumping assets in the order of minimum operation time/ highest priority/ minimum operation count / maximum idle time after operation first to maximum operation time/ least priority/ maximum operation count / minimum idle time based on the type of strategy run. In case of priority type, lower priority value corresponds to higher priority. If a balance strategy is selected, the order sequence is arranged by operation time. If a operation count strategy is selected the order sequence is arranged by operation count. If a idle time strategy is selected the order sequence is arranged by idle time after operation. If there are multiple pumping assets with same operation time or operation count or idle time, then the first connected pumping asset is considered to have a higher preference based on the strategy selected..

Function	Description
	If a priority strategy is selected, the order sequence is arranged by priority. If there are multiple pumping assets with the same priority, then for each set of same priority pumping assets, lesser operation time is considered as higher priority. Furthermore, in a set of pumping assets with same priority, if there are multiple pumping assets with same operation time, then the first connected pumping asset is considered to have a higher priority.
Switchover	If an undesirable condition is detected in an active pumping asset in the field, a Stop command is issued from the Pump Set to the inoperable pumping asset and the next available pumping asset in the order sequence is started. When the detected fault status of a pumping asset is reset, it automatically goes to Idle mode and waits for the command from the next calculation/ switching caused by the following factors: • Change in the number of requested pumping assets. • Detected failure of an active pump and switch over to an Idle one. • Recalculation is requested or forced. **NOTE:** Switchover does not apply if a user has paused/stopped/held a strategy during it's execution.
Operation	The Pump Set can be run in `Operator` or `Program` owner. Strategies are used to run and stop the Pump Set, and once it achieves the target (or target not achieved), the Strategy is completed and the equipment module state reflects the reality of the field. • Incorrect configuration or a detected failure condition will force the Pump Set to Stopped state. • If the Pump Set is in `Stopped` state, and the initial conditions are not satisfied (`INITCONDFAIL` is TRUE), then the Pump Set will go to Off state, and no command is possible, until the initial conditions are satisfied. • In Stopped state, the Pump Set can be started by selecting a strategy. Once a strategy is completed, if the Pump Set does not achieve the desired set point, it will take corrective action when a pumping asset becomes available for it to control. • Recalculation Strategy will capture the Priority/ operation time/operation count / idle time of the pumping assets, refresh the order sequence and evaluate the assets which need to run and those that can be stopped. • The pumping assets are started as per the order sequence. • The execution of Stop Strategy, or Abort command during a strategy execution, or a detected failure condition will stop the pumping assets in parallel.
Recalculation	Recalculation of the order sequence can be triggered either by the execution of the Recalculate strategy, or by the rising edge of the input pin. If the equipment module state was of Program type prior to recalculation, then the input pin `ACTIVEPUMPSSP` is considered for set point. After recalculating the order sequence, the Pump Set will try to achieve the set point (by starting the higher preference assets or stopping the lower preference assets). Once set point is achieved, Pump Set evaluates whether switch over is possible, by checking if an asset with higher preference in the order sequence is available and is idle, and pumping asset with lesser preference is available and active. This way, the Pump Set always tries to achieve the set point but does not exceeds it.
Detected Failure Condition	If a detected failure occurs then the Pump Set follows the Stop strategy execution matrix (Stopping to Stopped).

NOTE: For all the `MotorGP` instances connected to the Pump Set, the `ConfRearmEn` pins has to be `TRUE`. Also, the `TIMEOUT` value should be greater than 0 seconds.

NOTE: The `PUMPSETPATTERN` checks for the interlocks. If any asset is interlocked while in operation, the next available asset is taken into operation and on restoration of the interlock condition or on bypass ,the pumping asset starts again if the pumping asset has highest rank in the order sequence.

Strategy Selection and Functionality

Description

The given below table explains the selection of Strategy and its functionality.

Strategy	Strategy Number	Action
Program SP/Balance	1	The number of active pumps set point is defined by input pin `ACTIVEPUMPSSP`. This strategy is only enabled in case that the `ACTIVEPUMPSSP` input pin has been connected. In this strategy, the order sequence is determined by pump operation time.
Operator SP/Balance	2	The number of active pumps is defined by the `ACTIVEPUMPSSP` Input Parameter. In this strategy, the order sequence is determined by pump operation time.
Program SP/Priority	3	The number of active pumps is defined by input pin `ACTIVEPUMPSSP` of the DFB. This strategy is only enabled in case that the `ACTIVEPUMPSSP` input pin has been connected. In this strategy, the order sequence is determined by pump priority.
Operator SP/Priority	4	The number of active pumps is defined by the `ACTIVEPUMPSSP` input parameter. In this strategy, the order sequence is determined by pump priority.
Program SP/OperCount	5	The number of active pumps set point is defined by input pin `ACTIVEPUMPSSP`. This strategy is only enabled in case that the `ACTIVEPUMPSSP` input pin has been connected. In this strategy, the order sequence is determined by pump operation count.
Operator SP/OperCount	6	The number of active pumps is defined by the `ACTIVEPUMPSSP` Input Parameter. In this strategy, the order sequence is determined by pump operation count.
Program SP/Idletime	7	The number of active pumps set point is defined by input pin `ACTIVEPUMPSSP`. This strategy is only enabled in case that the `ACTIVEPUMPSSP` input pin has been connected. In this strategy, the order sequence is determined by pump idle time after operation.
Operator SP/Idletime	8	The number of active pumps is defined by the `ACTIVEPUMPSSP` Input Parameter. In this strategy, the order sequence is determined by pump idle time after operation.
Program SP/SwitchOver	9	The number of active pumps set point is defined by input pin `ACTIVEPUMPSSP`. This strategy is only enabled in case that the `ACTIVEPUMPSSP` input pin has been connected. In this strategy, the pumping assets switches over as per the switch over time defined by the user. **NOTE:** The recalculate strategy works only for the change of switchover time in the strategy and no switchover of pumpset expected.
Operator SP/SwitchOver	10	The number of active pumps is defined by the `ACTIVEPUMPSSP` Input Parameter. In this strategy, the pumping assets switches over as per the switch over time defined by the user. **NOTE:** The recalculate strategy works only for the change of switchover time in the strategy and no switchover of pumpset expected.
Recalculate	11	Forces the `PUMPSETPATTERN` DFB to refresh the order sequence and perform the recalculation function.
Stop Equipment	12	Stop operation of all the pumping assets simultaneously.

Disabling the Strategies Based on Equipment Module State

Description

For each equipment module state, the disabled strategies are shown below:

Equipment Module State	Code	Disabled Strategies
Off	0	1,2,3,4,5,6,7,8,9,10,11,12
Stopped	1	11,12
Stopping	10	1,2,3,4,5,6,7,8,9,10,11
Running Program SP/Balance	16	-
Running Operator SP/Balance	17	-
Running Program SP/Priority	18	-
Running Operator SP/Priority	19	-

Equipment Module State	Code	Disabled Strategies
Running Program SP/OperCount	20	-
Running Operator SP/OperCount	21	-
Running Program SP/IdleTime	22	-
Running Operator SP/OperCount	23	-
Running Program SP/SwitchOver	24	-
Operator SP/SwitchOver	25	-
Configuring	26	If strategy execution state is Idle, then all strategies are disabled. If not Idle, then all are disabled except the current strategy.
Running Externally	27	11,12

Input and Output Parameters

Description

The Input parameters are as shown below:

Variable Name	Parameter Number	Data Type	Description
IOPAR.ACTIVEPUMPSSP	1	INT	Number of Active Pumps
IOPAR.RESETOPERATIONTIME	2	BOOL	Reset Pump Cumulative Run Time
IOPAR.RESETOPERATIONCOUNT	3	BOOL	Reset Pump Operation Count
IOPAR.RESETIDLETIME	4	BOOL	Reset Pump Idle Time
IOPAR.RESETRUNTIME	5	DURATION	Reset Pump Run time
IOPAR.SWITCHOVERTIME	6	DURATION	Switch over time
IOPAR.PUMPPRIORITY[0]	7	INT	Priority of Pumping Asset 01
IOPAR.PUMPPRIORITY[1]	8	INT	Priority of Pumping Asset 02
IOPAR.PUMPPRIORITY[2]	9	INT	Priority of Pumping Asset 03
IOPAR.PUMPPRIORITY[3]	10	INT	Priority of Pumping Asset 04
IOPAR.PUMPPRIORITY[4]	11	INT	Priority of Pumping Asset 05
IOPAR.PUMPPRIORITY[5]	12	INT	Priority of Pumping Asset 06
IOPAR.PUMPPRIORITY[6]	13	INT	Priority of Pumping Asset 07
IOPAR.PUMPPRIORITY[7]	14	INT	Priority of Pumping Asset 08
IOPAR.PUMPPRIORITY[8]	15	INT	Priority of Pumping Asset 09
IOPAR.PUMPPRIORITY[9]	16	INT	Priority of Pumping Asset 10

The Output parameters are as shown below:

Variable Name	Parameter Number	Data Type	Description
IOPAR.EMDIAGSTATE	1	ENUM	Pump Set Diagnosis State
IOPAR.REQUESTEDPUMPS	2	INT	Number of requested Pumps
IOPAR.ACTIVEPUMPS	3	INT	Number of active pumps
IOPAR.PUMPOPERATIONTIME[0]/ IOPAR.PUMPOPERATIONCOUNT[0]/	4	DURATION /INT/	Cumulative Run Time of Pump 01/ Operation Count for Pump 01 / Idle Time for Pump 01/ Run Time of Pump 01

Variable Name	Parameter Number	Data Type	Description
IOPAR.PUMPIDLETIME[0]/IOPAR.PUMPRUNTIME[0]		DURATION/ DURATION	
IOPAR.ASSETDIAGSTATE[0]	5	INT	Diagnosis state of Pumping Asset 01 0: Normal, 1: Warning, 2: Failure
IOPAR.PUMPOPERATIONTIME[1]/ IOPAR.PUMPOPERATIONCOUNT[1]/ IOPAR.PUMPIDLETIME[1]/IOPAR.PUMPRUNTIME[1]	6	DURATION /INT/ DURATION/ DURATION	Cumulative Run Time of Pump 02/ Operation Count for Pump 02 / Idle Time for Pump 02/ Run Time of Pump 02
IOPAR.ASSETDIAGSTATE[1]	7	INT	Diagnosis state of Pumping Asset 02 0: Normal, 1: Warning, 2: Failure
IOPAR.PUMPOPERATIONTIME[2]/ IOPAR.PUMPOPERATIONCOUNT[2]/ IOPAR.PUMPIDLETIME[2]/IOPAR.PUMPRUNTIME[2]	8	DURATION /INT/ DURATION/ DURATION	Cumulative Run Time of Pump 03/ Operation Count for Pump 03 / Idle Time for Pump 03/ Run Time of Pump 03
IOPAR.ASSETDIAGSTATE[2]	9	INT	Diagnosis state of Pumping Asset 03 0: Normal, 1: Warning, 2: Failure
IOPAR.PUMPOPERATIONTIME[3]/ IOPAR.PUMPOPERATIONCOUNT[3]/ IOPAR.PUMPIDLETIME[3]/IOPAR.PUMPRUNTIME[3]	10	DURATION /INT/ DURATION/ DURATION	Cumulative Run Time of Pump 04/ Operation Count for Pump 04 / Idle Time for Pump 04/ Run Time of Pump 04
IOPAR.ASSETDIAGSTATE[3]	11	INT	Diagnosis state of Pumping Asset 04 0: Normal, 1: Warning, 2: Failure
IOPAR.PUMPOPERATIONTIME[4]/ IOPAR.PUMPOPERATIONCOUNT[4]/ IOPAR.PUMPIDLETIME[4]/IOPAR.PUMPRUNTIME[4]	12	DURATION /INT/ DURATION/ DURATION	Cumulative Run Time of Pump 05/ Operation Count for Pump 05 / Idle Time for Pump 05/ Run Time of Pump 05
IOPAR.ASSETDIAGSTATE[4]	13	INT	Diagnosis state of Pumping Asset 05 0: Normal, 1: Warning, 2: Failure
IOPAR.PUMPOPERATIONTIME[5]/ IOPAR.PUMPOPERATIONCOUNT[5]/ IOPAR.PUMPIDLETIME[5]/IOPAR.PUMPRUNTIME[5]	14	DURATION /INT/ DURATION/ DURATION	Cumulative Run Time of Pump 06/ Operation Count for Pump 06 / Idle Time for Pump 06/ Run Time of Pump 06
IOPAR.ASSETDIAGSTATE[5]	15	INT	Diagnosis state of Pumping Asset 06 0: Normal, 1: Warning, 2: Failure
IOPAR.PUMPOPERATIONTIME[6]/ IOPAR.PUMPOPERATIONCOUNT[6]/ IOPAR.PUMPIDLETIME[6]/IOPAR.PUMPRUNTIME[6]	16	DURATION /INT/ DURATION/ DURATION	Cumulative Run Time of Pump 07/ Operation Count for Pump 07 / Idle Time for Pump 07/ Run Time of Pump 07
IOPAR.ASSETDIAGSTATE[6]	17	INT	Diagnosis state of Pumping Asset 07 0: Normal, 1: Warning, 2: Failure
IOPAR.PUMPOPERATIONTIME[7]/ IOPAR.PUMPOPERATIONCOUNT[7]/ IOPAR.PUMPIDLETIME[7]/IOPAR.PUMPRUNTIME[7]	18	DURATION /INT/ DURATION/ DURATION	Cumulative Run Time of Pump 08/ Operation Count for Pump 08 / Idle Time for Pump 08/ Run Time of Pump 08
IOPAR.ASSETDIAGSTATE[7]	19	INT	Diagnosis state of Pumping Asset 08 0: Normal, 1: Warning, 2: Failure
IOPAR.PUMPOPERATIONTIME[8]/ IOPAR.PUMPOPERATIONCOUNT[8]/ IOPAR.PUMPIDLETIME[8]/IOPAR.PUMPRUNTIME[8]	20	DURATION /INT/ DURATION/ DURATION	Cumulative Run Time of Pump 09/ Operation Count for Pump 09 / Idle Time for Pump 09/ Run Time of Pump 09
IOPAR.ASSETDIAGSTATE[8]	21	INT	Diagnosis state of Pumping Asset 09 0: Normal, 1: Warning, 2: Failure
IOPAR.PUMPOPERATIONTIME[9]/ IOPAR.PUMPOPERATIONCOUNT[9]/ IOPAR.PUMPIDLETIME[9]/IOPAR.PUMPRUNTIME[9]	22	DURATION /INT/ DURATION/ DURATION	Cumulative Run Time of Pump 10/ Operation Count for Pump 10 / Idle Time for Pump 10/ Run Time of Pump 10
IOPAR.ASSETDIAGSTATE[9]	23	INT	Diagnosis state of Pumping Asset 10: Normal, 1: Warning, 2: Failure

Input Parameters Applicable For Each Strategy

Description

The below table provides the information about the input parameters that are applicable for each strategy.

Strategy	Input Parameters Applicable
Program SP/Balance	2, 3, 4, 5
Operator SP/Balance	1, 2, 3, 4, 5
Program SP/Priority	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
Operator SP/Priority	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
Program SP/Operation Count	2, 3, 4, 5
Operator SP/Operation Count	1, 2, 3, 4, 5
Program SP/Idletime	2, 3, 4
Operator SP/Idletime	1, 2, 3, 4
Program SP/Idletime	2, 3, 4
Operator SP/Idletime	1, 2, 3, 4
Program SP/SwitchOver	2, 3, 4, 5, 6
Operator SP/SwitchOver	1, 2, 3, 4, 5, 6
Recalculate	2, 3, 4, 5, 6
Stop Equipment	2, 3, 4, 5, 6

Pump Set Equipment Module State Machine

Description

The below given table provides the list of applicable strategies for each Equipment Module State, the transient state when the strategy is started, and the final equipment module state once the strategy is completed:

Initial EM State	Strategy Command	Transient EM State	Final EM State
Off	No action possible	-	Stopped
			Running Externally
Stopped	Program SP/Balance	Configuring	Running Program SP/Balance
	Operator SP/Balance	Configuring	Running Operator SP/Balance
	Program SP/Priority	Configuring	Running Program SP/Priority
	Operator SP/Priority	Configuring	Running Operator SP/Priority
	-	-	Off
	-	-	Running Externally
	Program SP/OperCount	Configuring	Running ProgramSP/OperCount
	Operator SP/OperCount	Configuring	Running OperatorSP/OperCount
	Program SP/IdleTime	Configuring	Running ProgramSP/IdleTime
	Operator SP/dleTime	Configuring	Running OperatorSP/IdleTime
	Program SP/SwitchOver	Configuring	Running ProgramSP/SwitchOver
	Operator SP/SwitchOver	Configuring	Running OperatorSP/SwitchOver

Initial EM State	Strategy Command	Transient EM State	Final EM State
• Running Program SP/Balance • Running Operator SP/Balance • Running Program SP/Priority • Running Operator SP/Priority • Running Program SP/ OperCount • Running Operator SP/ OperCount • Running Program SP/Idletime • Running Operator SP/Idletime • Running Program SP/ SwitchOver • Running Operator SP/ SwitchOver	`Recalculate`	Configuring	RunningX*
	`Program SP/Balance`	Configuring	Running Program SP/Balance
	`Operator SP/Balance`	Configuring	Running Operator SP/Balance
	`Program SP/Priority`	Configuring	Running Program SP/Priority
	`Operator SP/Priority`	Configuring	Running Operator SP/Priority
	`Program SP/OperCount`	Configuring	Running Program SP/OperCount
	`Operator SP/OperCount`	Configuring	Running Operator SP/OperCount
	`Program SP/Idletime`	Configuring	Running Program SP/Idletime
	`Operator SP/Idletime`	Configuring	Running Operator SP/Idletime
	`Stop Equipment`	Stopping	Stopped
	-	-	Running Externally
	`Program SP/SwitchOver`	Configuring	Running ProgramSP/SwitchOver
	`Operator SP/SwitchOver`	Configuring	Running OperatorSP/SwitchOver
Configuring	No action possible	-	-
Stopping	No action possible	-	-
Running Externally	`Program SP/Balance`	Configuring	Running Program SP/Balance
	`Operator SP/Balance`	Configuring	Running Operator SP/Balance
	`Program SP/Priority`	Configuring	Running Program SP/Priority
	`Operator SP/Priority`	Configuring	Running Operator SP/Priority
	`Program SP/OperCount`	Configuring	Running Program SP/OperCount
	`Operator SP/OperCount`	Configuring	Running Operator SP/OperCount
	`Program SP/Idletime`	Configuring	Running Program SP/Idletime
	`Operator SP/Idletime`	Configuring	Running Operator SP/Idletime
	-	-	Stopped
	`Program SP/SwitchOver`	Configuring	Running ProgramSP/SwitchOver
	`Operator SP/SwitchOver`	Configuring	Running OperatorSP/SwitchOver
*: X represents initial Equipment Module State			

Pump Set Equipment Module Diagnosis State

Description

The Pump Set equipment module diagnosis state is evaluated as given below:

- `Normal`: State is valid if below condition is true:
  - There are no detected failure and alert conditions.
- `Warning`: State is valid if any of below condition is true:
  - When total number of active pumping assets are more than number of assets requested by pump set pattern.
  - If stop strategy could not stop all the pumping assets.
  - If requested number of assets is equal to fully active assets and there is a problem in any pumping asset.

- `Failure`: State is valid if any of below condition is true:
  - When there is an incorrect configuration.
  - When there is a detected failure condition.
  - When the number of active pumping assets is lesser than number of pumping assets required by pump set pattern.

  NOTE: The Pump Set equipment module diagnosis state is not evaluated when the Pump Set is in transient state (Configuring or Stopping) to avoid unnecessary alert and detected failure generation. However, the Pump Set detected failure and incorrect configuration are always evaluated.

Pump Set Pumping Asset Diagnosis State

Description

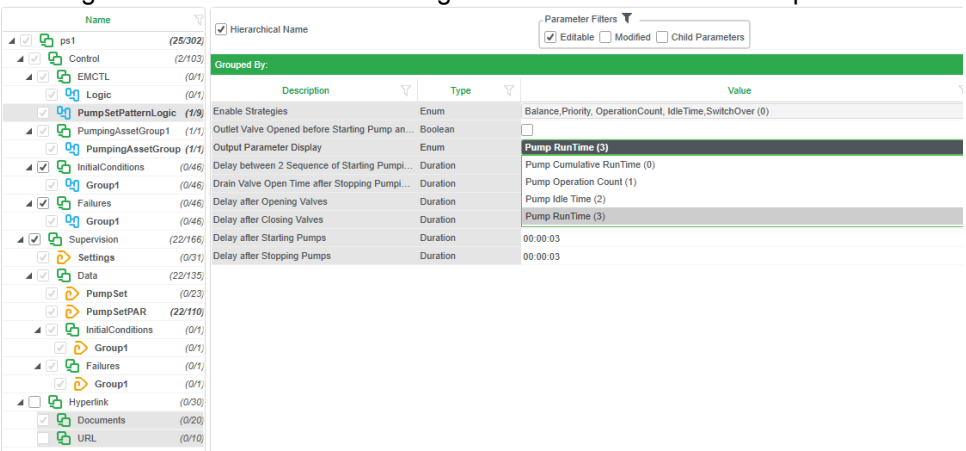
Each individual pumping asset has its own diagnosis state and each of them are evaluated as given below:

- `Normal`: The set of Valves and Pump is operating properly.
- `Warning`: The set of Valves and Pump caused a problem during its last operation. The last time that the algorithm tried to execute the complete sequence for starting/stopping the Pumping Asset for some reason (for example, the operating mode was not correct, it triggered an alarm, the confirmations were not reached, etc.). Switching from alert to a detected failure does not reset the alert condition, the only way that the alert is reset is when the same Pumping Asset is operated again for starting or stopping it.
- `Failure`: Any one of the Valves or Pump is in inoperable condition (The `FAILD`, `ALARM`, `REARMR`, `OOS` or `ILCKD` value is TRUE).

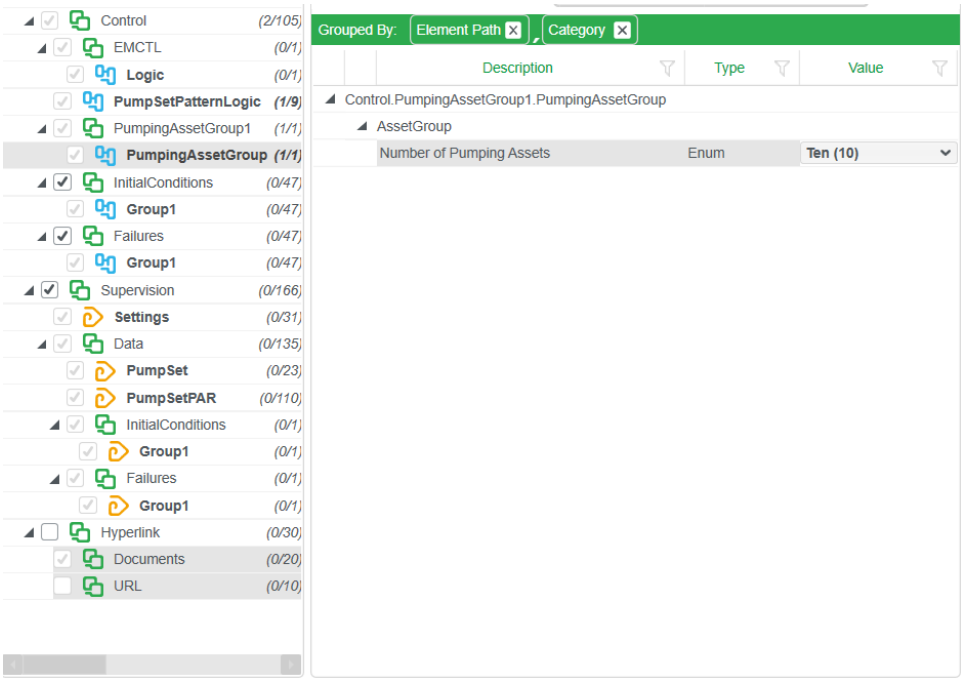
Modifying the PumpSetPattern Template

Description

The `PUMPSETPATTERN` can handle up to 10 pumping assets and by default can be connected to 2 pumping assets also the user has the facility to select the required output parameter (cumulative run time / operation count / idle time / run time) to be displayed. The output parameters which are not selected for display will be generated as variables allowing the user to use the data if required.



The user can select the required number of pumping assets from 2 to 10 with help of parameter named `Number of Pumping Assets`.

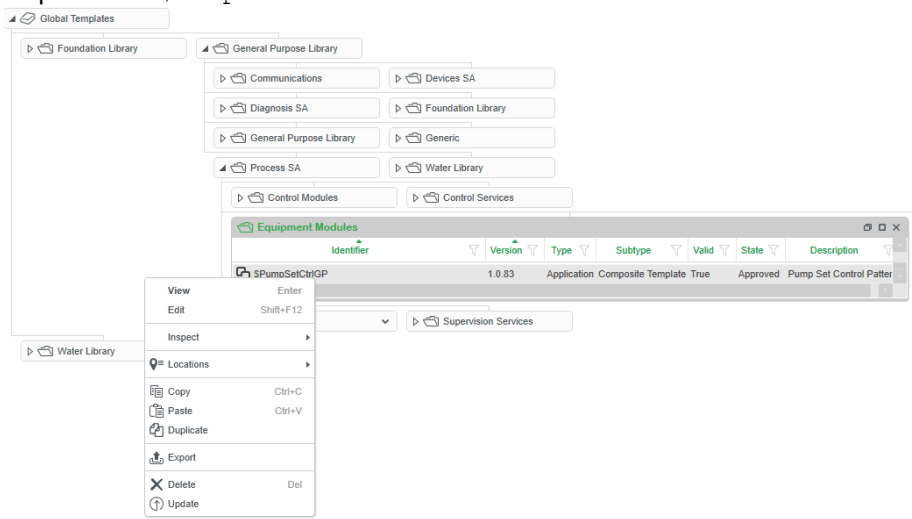


Perform the following steps to connect the 15 Pumping assets:

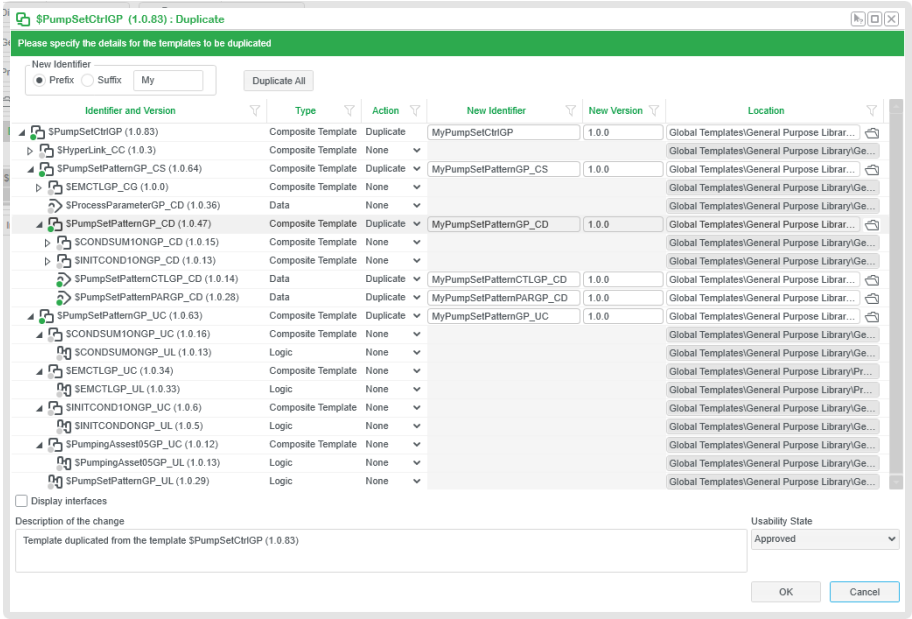
- Modify `$PumpingAsset05GP_UL`.
- Update `$PumpingAsset05GP_UC`.
- Add a new instance of `$PumpingAsset05GP_UC` in `$PumpSetPatternGP_UC`
- Modify `$PumpSetPatternPARGP_CD`
- Update `$PumpSetPatternGP_CD`
- Update `$PumpSetPatternGP_CS`
- Update `$PumpSetCtrlGP`

Perform the following steps to modify the `PumpSetCtrlGP` Template to handle 15 pumping assets:

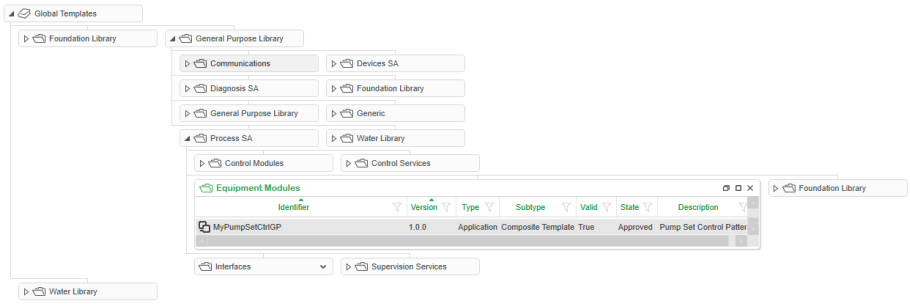
1. Duplicate the`$PumpSetCtrlGP`.



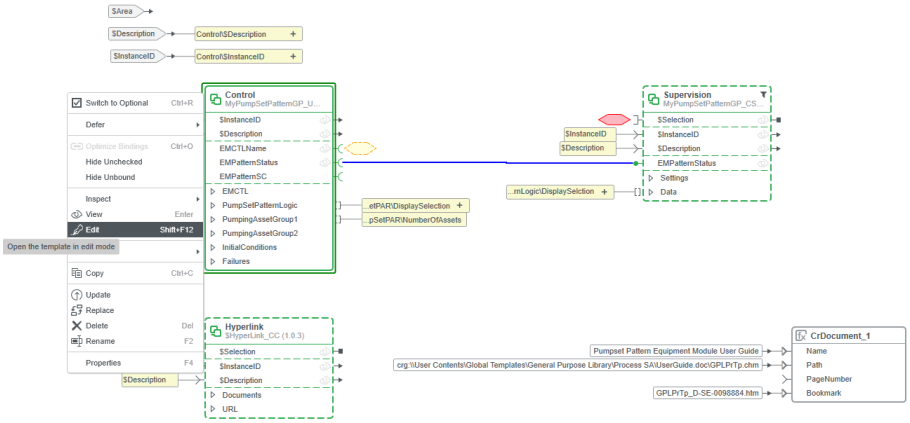
2. Name the new template with a recognizable name.



Find the new template under the equipment module.

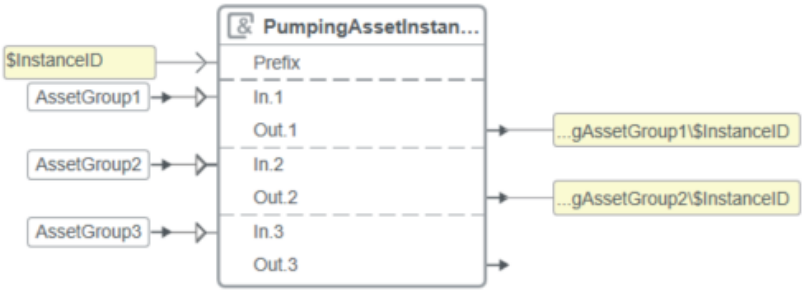


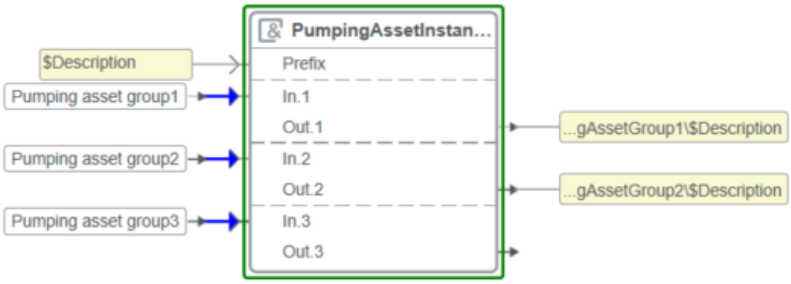
3. Open the Duplicated template and go to edit mode of the Control composite.



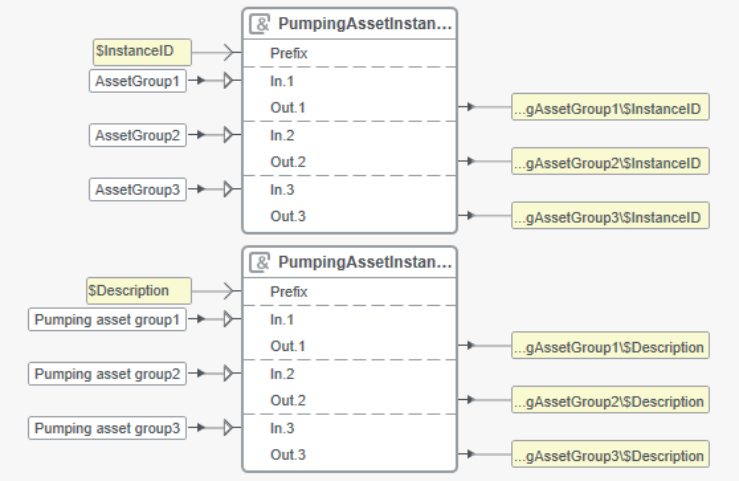
4. Add a new UC of `$PumpingAssest05GP` and give the value for group number as three, make the create condition with greater than or equal to 10 and link as shown in below image. Perform step 5 and step 6 before saving the template.

5. Extend the `PumpingAssetInstanceName(MConcat)` and `PumpingAssetInstanceDescription` to 3 and add an input parameter called `AssetGroup3`.

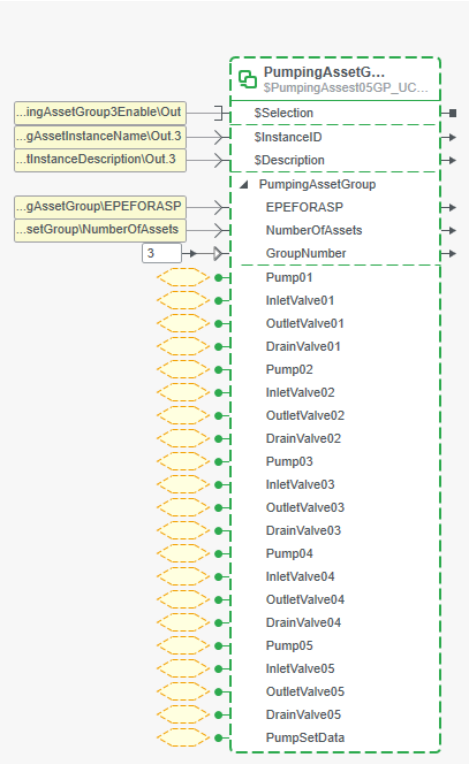




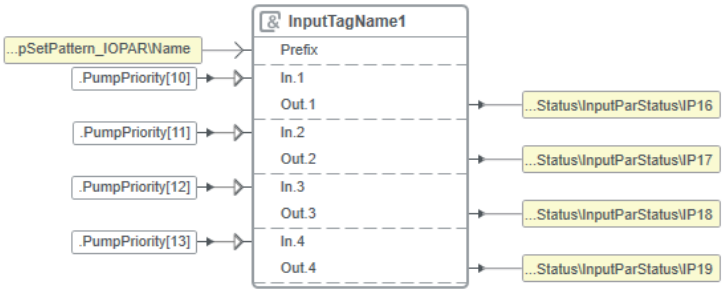
6. Link the instance name and description to the newly added `$PumpingAsset05_UC`.

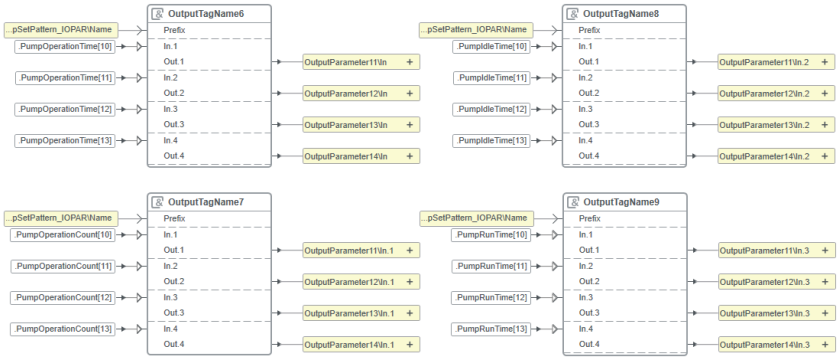


After linking, the newly added `$PumpingAssest05GP` appears as shown below.



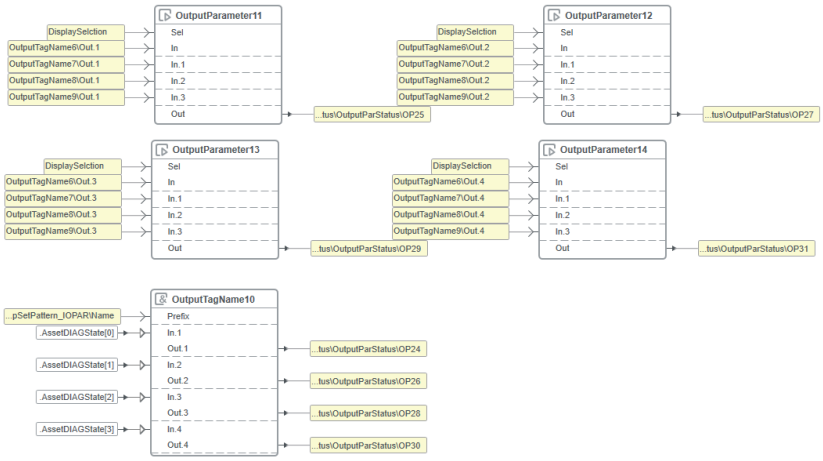
7. Edit `$PumpSetPatternGP_UL` to add data of four more pumping assets.





8. Add the logic for generation of deselected parameters. Link the newly added deselected parameters to EM `PumpsetPattern` interface.

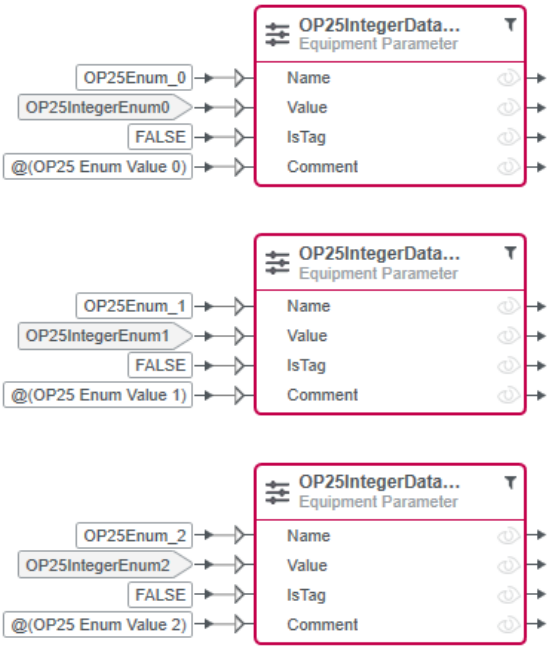
9. Link the newly added parameters to the`EMParStatus` Interface.



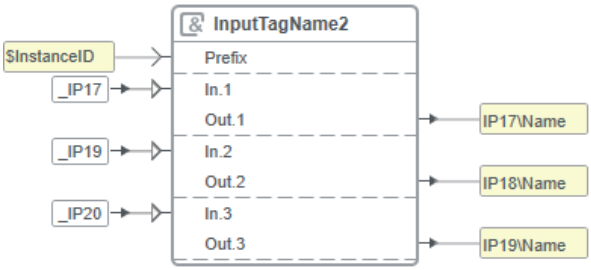
Save the template and update the logic template in the composite template.

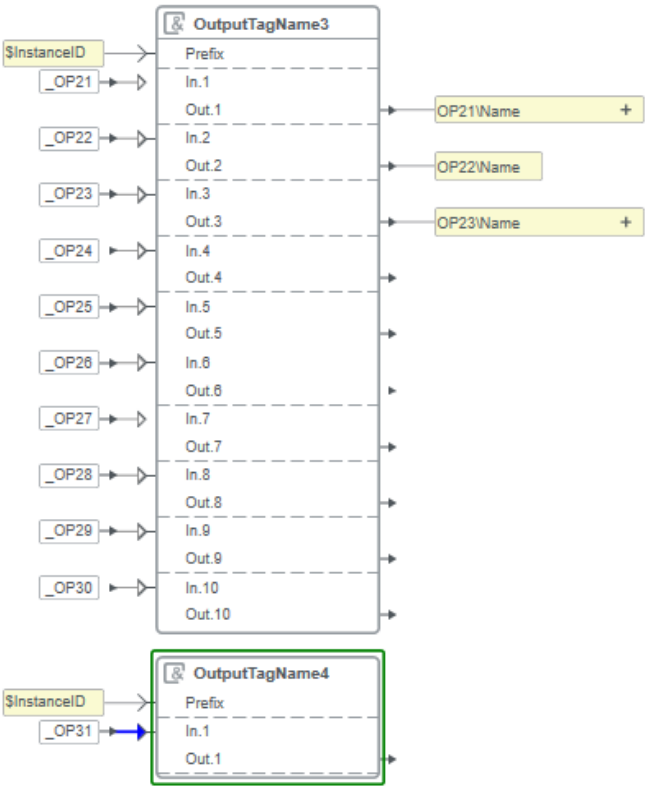
10. Edit `$PumpSetPARGP_CD` or its duplicated template in the supervision to generate newly added tags, alarms and enumerations.

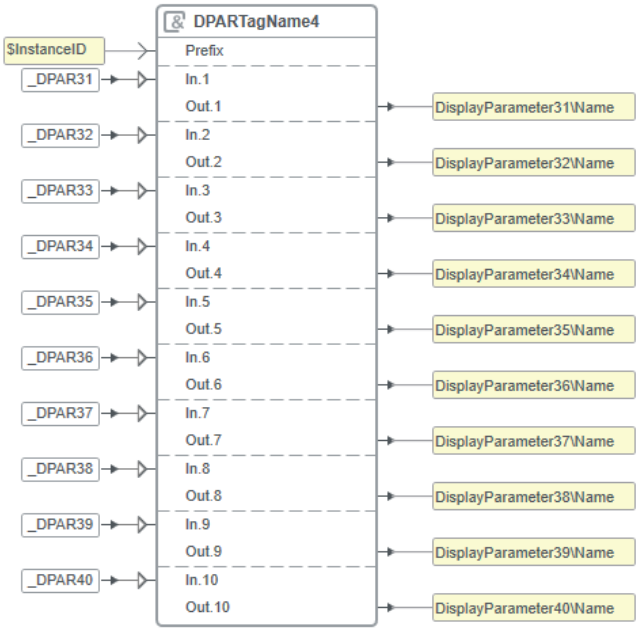
    Replicate the following with change of names as OP25, OP27, OP29, OP31 instead of OP23 for the input values, parameters and the Equipment Parameter names. As shown below replicate the same for OP27, OP29, OP31.



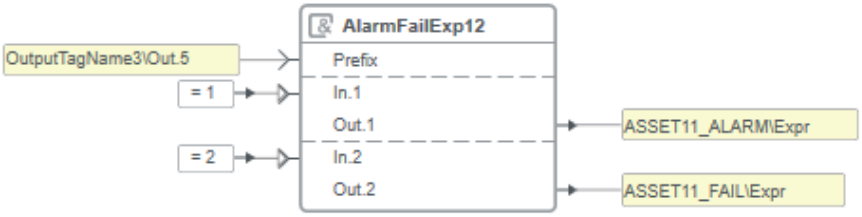
11. Extend Output Tag names, Input tag Names and connect to the newly create input and output variable tags.

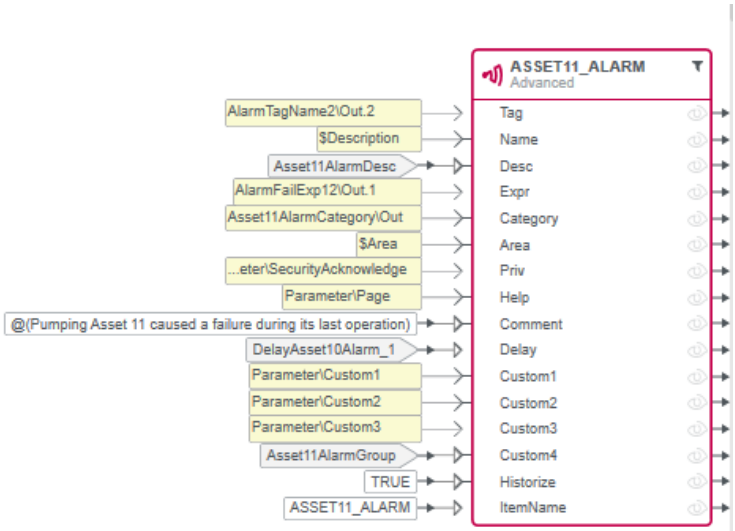


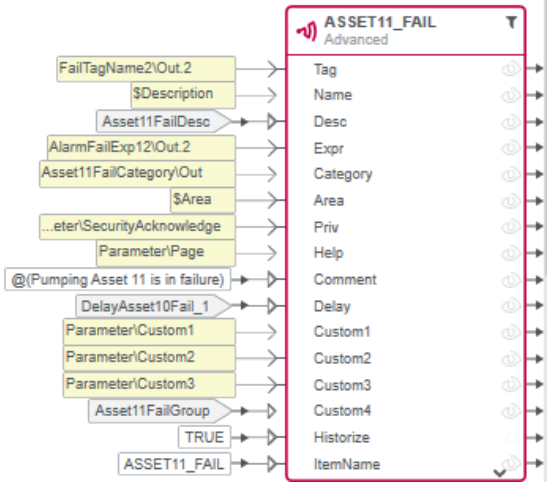

OutputTagName3
$InstanceID
Prefix
_OP21
In.1
Out.1
OP21\Name
_OP22
In.2
Out.2
OP22\Name
_OP23
In.3
Out.3
OP23\Name
_OP24
In.4
Out.4
_OP25
In.5
Out.5
_OP26
In.6
Out.6
_OP27
In.7
Out.7
_OP28
In.8
Out.8
_OP29
In.9
Out.9
_OP30
In.10
Out.10
OutputTagName4
$InstanceID
Prefix
_OP31
In.1
Out.1



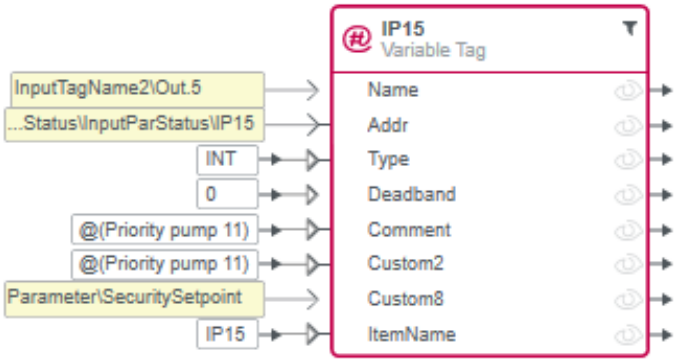
12. Create Asset alarms 11, 12, 13 and 14 as shown below. Replicate the same for 12, 13, 14.



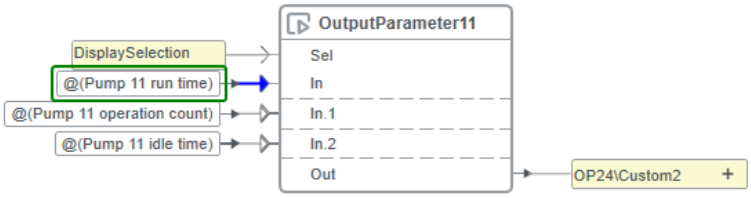

ASSET11_ALARM
Advanced
AlarmTagName2\Out.2
$Description
Asset11AlarmDesc
AlarmFailExp12\Out.1
Asset11AlarmCategory\Out
$Area
...eter\SecurityAcknowledge
Parameter\Page
@(Pumping Asset 11 caused a failure during its last operation)
DelayAsset10Alarm_1
Parameter\Custom1
Parameter\Custom2
Parameter\Custom3
Asset11AlarmGroup
TRUE
ASSET11_ALARM
Tag
Name
Desc
Expr
Category
Area
Priv
Help
Comment
Delay
Custom1
Custom2
Custom3
Custom4
Historize
ItemName

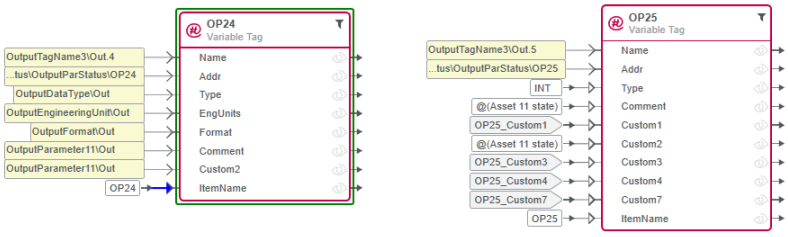

ASSET11_FAIL
Advanced
FailTagName2\Out.2
$Description
Asset11FailDesc
AlarmFailExp12\Out.2
Asset11FailCategory\Out
$Area
...eter\SecurityAcknowledge
Parameter\Page
@(Pumping Asset 11 is in failure)
DelayAsset10Fail_1
Parameter\Custom1
Parameter\Custom2
Parameter\Custom3
Asset11FailGroup
TRUE
ASSET11_FAIL
Tag
Name
Desc
Expr
Category
Area
Priv
Help
Comment
Delay
Custom1
Custom2
Custom3
Custom4
Historize
ItemName

13. Create Variable tags for Pump Priority, Pump Asset State, Selected display parameter and non selected parameters for Pump 11 , Pump 12, Pump 13 and Pump 14.
    - For Input tags: Replicate the same for IP15, IP16, IP17, IP18.

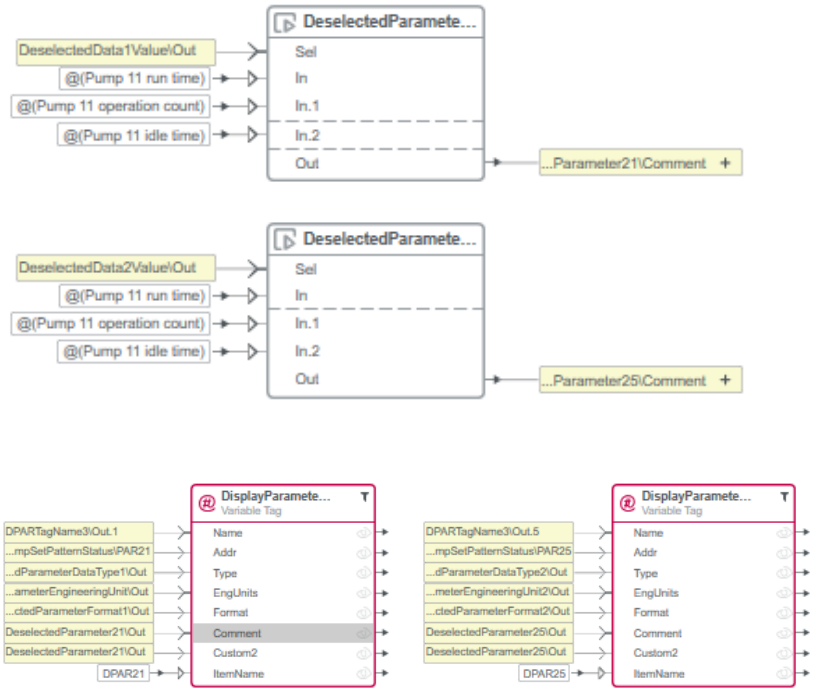


    - For Output tags : Replicate the same for OP26, OP27, OP28, OP29, OP30, OP31.



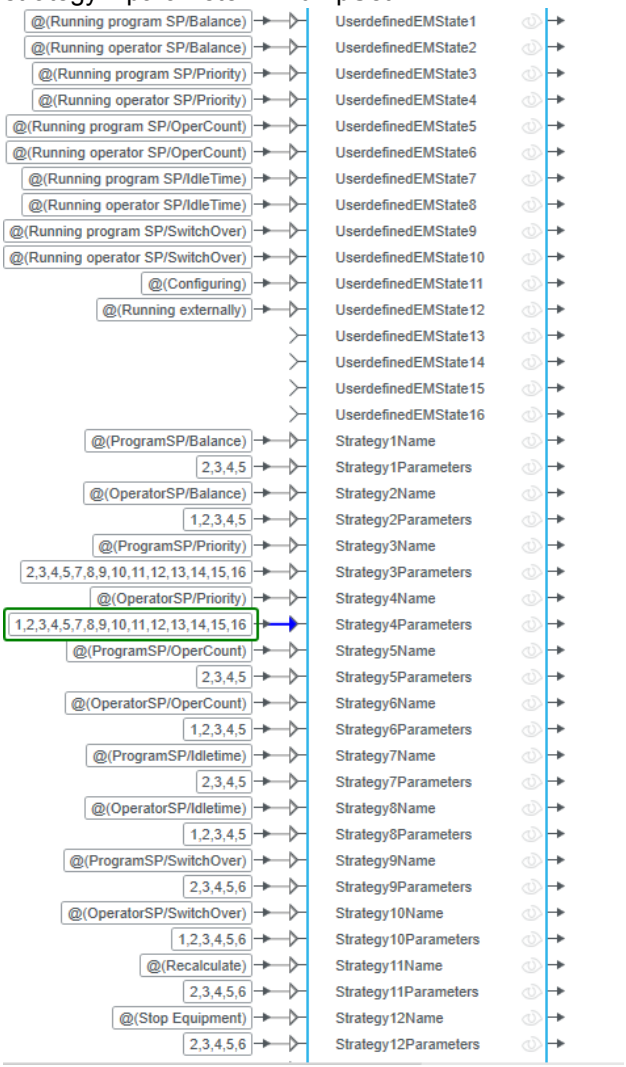


    - For non selected parameters : Replicate the same for DPAR31, DPAR32, DPAR24, DPAR26, DPAR27, DPAR28



Save the template.

14. In `$PumpSetPatternGP_CD` or its duplicate, Update the strategy 3 and strategy 4 parameter in PumpSet.



15. Save the template. Update the newly created templates, control and supervision part.

Pump Set

What's in This Chapter

Template 785
Control — PUMPSETPATTERNGP 791
Control — PUMPINGASSET05GP 797
Supervision 799

Overview

This chapter describes the components that provide the functionality of template, control services, and Supervision functions of the pump set.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$PumpSetCtrlGP` template.

Description

General Description

The `$PumpSetCtrlGP` template is used to monitor and manage the Pump Set operation. It is provided with the configuration of 5 pumping assets. Each asset includes one Pump and three Valves (Inlet, Outlet and Drain).

Function Description

The main functions of the template are described in the following table:

Function	Description
Incorrect Configuration	Determines the total number of pumps connected to the PumpingAsset05GP from the connections to the pump input pins of the DFB. If the pump connections are not in order, it detects an incorrect configuration. For example, If `PUMP01`, `PUMP02`, `PUMP04` input pins are connected, the incorrect configuration is detected since the pumps are not connected sequentially. When there is a incorrect configuration is detected, `PUMPSETPATTERN` equipment is not available for operation.
Order Sequence	The order sequence arranges the pumping assets in the order of minimum operation time/ highest priority/ minimum operation count / maximum idle time after operation first to maximum operation time/ least priority/ maximum operation count / minimum idle time last based on the type of strategy run. In case of priority type, lower priority value corresponds to higher priority. If a balance strategy is selected, the order sequence is arranged by operation time. If a operation count strategy is selected the order sequence is arranged by operation count. If a idle time strategy is selected the order sequence is arranged by idle time after operation. If there are multiple pumping assets with same operation time or operation count or idle time, then the first connected pumping asset is considered to have a higher preference based on the strategy selected.. If a priority strategy is selected, the order sequence is arranged by priority. If there are multiple pumping assets with the same priority, then for each set of same priority pumping assets, lesser operation time is considered as higher priority. Furthermore, in a set of pumping assets with same priority, if there are multiple pumping assets with same operation time, then the first connected pumping asset is considered to have a higher priority.
Switchover	If an undesirable condition is detected in an active pumping asset in the field, a Stop command is issued from the Pump Set to the inoperable pumping asset and the next available pumping asset in the order sequence is started.

Function	Description
	When the detected fault status of a pumping asset is reset, it automatically goes to Idle mode and waits for the command from the next calculation/ switching caused by the following factors: • Change in the number of requested pumping assets. • Detected failure of an active pump and switch over to an Idle one. • Recalculation is requested or forced. **NOTE:** Switchover does not apply if a user has paused/stopped/held a strategy during it's execution.
Operation	The Pump Set can be run in `Operator` or `Program` owner. Strategies are used to run and stop the Pump Set, and once it achieves the target (or target not achieved), the Strategy is completed and the equipment module state reflects the reality of the field. • Incorrect configuration or a detected failure condition will force the Pump Set to Stopped state. • If the Pump Set is in `Stopped` state, and the initial conditions are not satisfied (`INITCONDFAIL` is TRUE), then the Pump Set will go to Off state, and no command is possible, until the initial conditions are satisfied. • In Stopped state, the Pump Set can be started by selecting a strategy. Once a strategy is completed, if the Pump Set does not achieved the desired set point, it will take corrective action when a pumping asset becomes available for it to control. • Recalculation Strategy will capture the Priority/ operation time/operation count / idle time of the pumping assets, refresh the order sequence and evaluate the assets which need to run and those that can be stopped. • The pumping assets are started as per the order sequence. • The execution of Stop Strategy, or Abort command during a strategy execution, or a detected failure condition will stop the pumping assets in parallel.
Recalculation	Recalculation of the order sequence can be triggered either by the execution of the Recalculate strategy, or by the rising edge of the input pin. If the equipment module state was of Program type prior to recalculation, then the input pin `ACTIVEPUMPSSP` is considered for set point. After recalculating the order sequence, the Pump Set will try to achieve the set point (by starting the higher preference assets or stopping the lower preference assets). Once set point is achieved, Pump Set evaluates whether switch over is possible, by checking if an asset with higher preference in the order sequence is available and is idle, and pumping asset with lesser preference is available and active. This way, the Pump Set always tries to achieve the set point but does not exceeds it.
Detected Failure Condition	If a detected failure occurs then the Pump Set follows the Stop strategy execution matrix (Stopping to Stopped).

NOTE: For all the `MotorGP` instances connected to the Pump Set, the `ConfRearmEn` pins has to be `TRUE`. Also, the `TIMEOUT` value should be greater than 0 seconds.

NOTE: The `PUMPSETPATTERN` checks for the interlocks. If any asset is interlocked while in operation, the next available asset is taken into operation and on restoration of the interlock condition or on bypass ,the pumping asset starts again if the pumping asset has highest rank in the order sequence.

Parameters

Parameters

The `$PumpSetCtrlGP` template defines operation of pump set. Alternative pump set input facets enable the modification of the pump set that are available.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Disable auto reset	`Boolean`	False	–
	Enable strategies	`Enum`	Balance, Priority, OperationCount, IdleTime (0)	–

Section	Parameter description	Type	Default	Additional remarks
	Outlet valve opened before starting pump and closed after stopping pump	Boolean	False	–
	Output Parameter Display	Enum	Pump Operation Time (0)	–
Time	Delay between 2 sequence of starting pumping asset	Duration	00:00:03	–
	Drain valve open time after stopping pumping asset		00:00:05	–
	Delay after opening valves		00:00:03	–
	Delay after closing valves		00:00:03	–
	Delay after starting pumps		00:00:03	–
	Delay after stopping pumps		00:00:03	–
AssetGroup	Number of Pumping Assets	Enum	Two (2)	–
Description	**InitialConditions**			
	Condition 1 to 15	String	–	–
	Group description		@(Initial condi-tions)	–
	FailureConditions			
	Condition 1 to 15	String	–	–
	Group description		@ (Failures)	–
Reset Required	**InitialConditions**			
	Condition 1 to 15	Boolean	False	–
	FailureConditions			
	Condition 1 to 15	Boolean	False	–
Disable Bypass	**InitialConditions**			
	Condition 1 to 15	Boolean	False	–
	FailureConditions			
	Condition 1 to 15	Boolean	False	–

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Operation	Normal Owner	Enum	Program (1)	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	Set owner	Enum	Operator (1)	Privilege to change owner. This parameter is used in Operator Tab, page 800.
	Set setpoint		Operator (1)	Privilege to set setpoint. This parameter is used in Operator Tab, page 800.
	Set reset		Operator (Confirmed) (10)	Privilege to reset. This parameter is used in Operator Tab, page 800.

Section	Parameter description	Type	Default	Additional remarks
	Set alarm acknowledge		Operator (1)	Privilege to set alarm acknowledge. This parameter is used in Alarm Tab, page 67.
	Set bypass conditions		Supervisor (2)	Privilege to change owner.
	Set semi mode		Operator (1)	Privilege to set semi mode. This parameter is used in Operator Tab, page 800.
	Set manual mode			Privilege to set manual mode. This parameter is used in Operator Tab, page 800.
	Set next step			Privilege to set next step. This parameter is used in Operator Tab, page 800.
	Set scroll up			Privilege to set scroll up. This parameter is used in Operator Tab, page 800.
	Set scroll down			Privilege to set scroll down. This parameter is used in Operator Tab, page 800.
	Set start			Privilege to set start. This parameter is used in Operator Tab, page 800.
	Set hold			Privilege to set hold. This parameter is used in Operator Tab, page 800.
	Set restart			Privilege to set restart. This parameter is used in Operator Tab, page 800.
	Set stop			Privilege to set stop. This parameter is used in Operator Tab, page 800.
	Set reset		Operator (Confirmed) (10)	Privilege to reset. This parameter is used in Operator Tab, page 800.
	Set pause		Operator (1)	Privilege to set pause. This parameter is used in Operator Tab, page 800.
	Set resume			Privilege to set resume. This parameter is used in Operator Tab, page 800.
Security	Set abort	`Enum`	Operator (1)	Privilege to set abort. This parameter is used in Operator Tab, page 800.
	Set strategy			Privilege to set strategy. This parameter is used in Operator Tab, page 800.

Asset and Display Parameters

For more details, refer to the topic describing the parameters of asset and display, page 43.

Data

The table describes the data parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Disable start command button	`Boolean`	False	–
	Disable hold command button			–
	Disable pause command button			–
	Disable reset command button			–
	Disable stop command button			–
	Disable abort command button			–
	Disable restart command button			–
Alarm	Held for *error* severity	`Enum`	Low (4000)	–

Section	Parameter description	Type	Default	Additional remarks
	Held for *error* description		@(Held For Error)	–
	Held for *error* group	`Enum`	*Failure*	–
Event	Enable log	`Boolean`	False	Enable event logging for asset status.
Historize	State	`Boolean`	False	–
	Current step description			–
	Step description			–
	Next step description			–
	Transition description			–
	Disable strategy			–
	Equipment module state			–
	Equipment output status			–
	Sequence equipment module mode			–
	Equipment module strategy			–
	Command			–
	Elapsed time			–
Alarm	Equipment module disable	`Boolean`	True	–
	Equipment module discription	`String`	@(EM Alarm Description)	–
	Equipment module severity	`Enum`	Low (4000)	–
	Equipment module group	`String`	*Failure*	–
	Equipment module delay	`Dura-tion`	00:00:00	–
	Asset 01 disable	`Boolean`	True	–
	...			–
	Asset 10 disable			–
	Asset 01 description	`String`	@(Asset 01 Alarm Description)	–
	...		...	–
	Asset 10 description		@(Asset 10 Alarm Description)	–
	Asset 01 severity	`Enum`	Low (4000)	–
	...			–
	Asset 10 severity			–
	Asset 01 group	`String`	*Failure*	–
	...			–
	Asset 10 group			–
	Asset 01 delay	`Dura-tion`	00:00:00	–
	...			–
	Asset 10delay			–
Alarm	Equipment module failure disable	`Boolean`	True	–

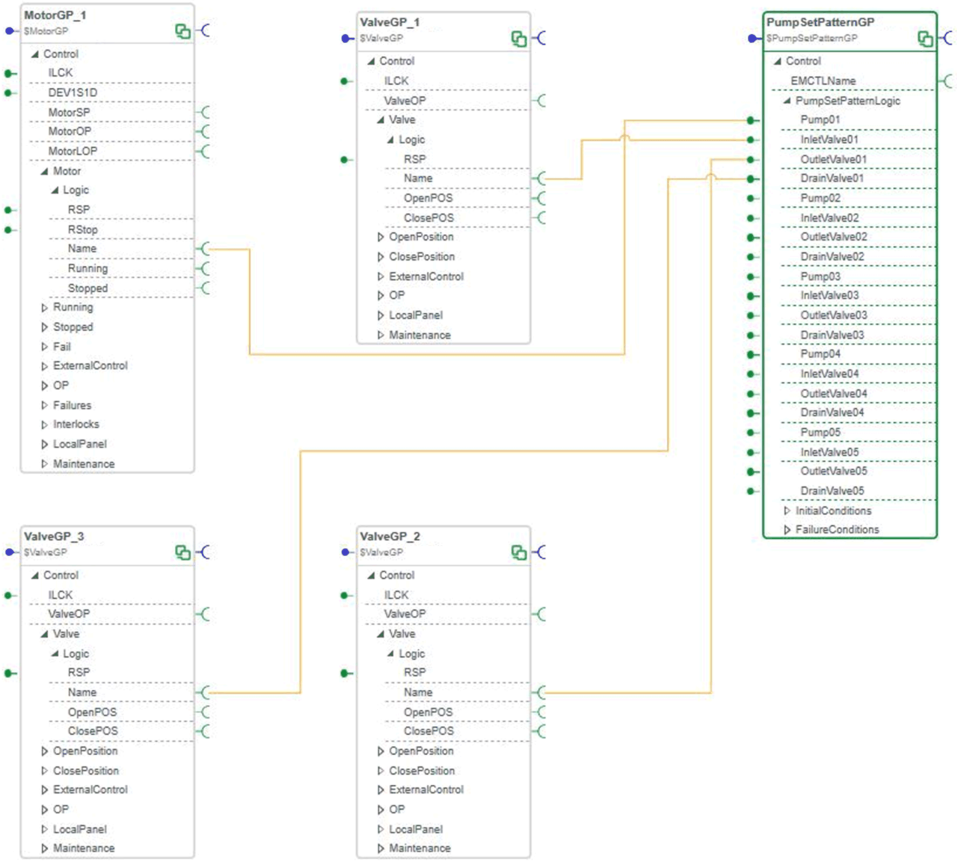
Section	Parameter description	Type	Default	Additional remarks
	Equipment module failure discription	String	@(EM Alarm Description)	–
	Equipment module failure severity	Enum	Low (4000)	–
	Equipment module failure group	String	*Failure*	–
	Equipment module failure delay	Duration	00:00:00	–
	Asset 01 failure disable	Boolean	True	–
	...			–
	Asset 10 failure disable			–
	Asset 01 failure discription	String	@(Asset 01 *Failure* Description)	–
	...		...	–
	Asset 10 failure discription		@(Asset 10 *Failure* Description)	–
	Asset 01 failure severity	Enum	Low (4000)	–
	...			–
	Asset 10 failure severity			–
	Asset 01 failure group	String	*Failure*	–
	...			–
	Asset 10 failure group			–
	Asset 01 failure delay	Duration	00:00:00	–
	...			–
	Asset 10 failure delay			–
Historize	**InitalConditionTags** and **Failure Conditions Tags**			
	Conditions	Boolean	False	–

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$PumpSetCtrlGP` (also applicable for `$PumpSetCtrlCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



For interface identifiers, refer to the control module template interfaces, page 49.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control — PUMPSETPATTERNGP

Overview

This chapter describes the `PUMPSETPATTERNGP` DFB.

Description

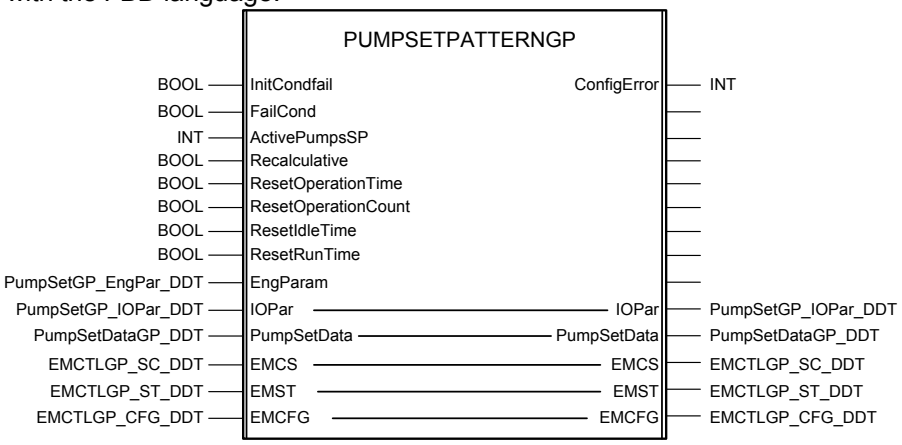
General

The `PUMPSETPATTERNGP` DFB can be used to run common pumping applications like Sump management, Transfer and Circulation pump, etc. The maximum number of pumping assets supported for operation is five. The number of pumping assets can be modified according to the users application requirement.

DFB Representation

Representation

The DFB that is used in the program has the following aspect at the section level. You can use it in any of the programming languages, although it is designed for use with the FBD language.



Inputs

Input Parameter Description

Parameter	Type	Description
`InitCondfail`	`BOOL`	Detected failure in initial condition.
`FailCond`	`BOOL`	Detected failure condition.
`ActivePumpsSP`	`INT`	Remote requested number of active pumps. Minimum value considered by the `PUMPSETPATTERNGP` DFB is 1 and the maximum value considered is the number of configured pumps.
`Recalculative`	`BOOL`	Recalculate ON/OFF status of pumps based on current equipment module state.
`ResetOperation-Time`	`BOOL`	Reset the operation time of all the pumps.
`ResetOperation-Count`	`BOOL`	Reset the operation count of all the pumps.
`ResetIdleTime`	`BOOL`	Reset the idle time of all the pumps.
`ResetRunTime`	`BOOL`	Reset the run time of all running pumps.
`EngParam`	`PumpSetGP_EngPar_DDT`	Pump set engineering parameters. **NOTE:** If this input pin is not connected, then the following default values are considered by the DFB: • `SwitchOnDelay`: 3s • `DrainingTime`: 5s • `DelayAfterValveOpen`: 3s • `DelayAfterValveClose`: 3s • `DelayAfterPumpStart`: 3s • `DelayAfterPumpStop`: 3s • `EnabledStrategies`: 0 • `OutValveFirst`: FALSE

PumpSetGP_EngPar_DDT Type

Name	Type	Description
SwitchOnDelay	TIME	Delay between the end of the starting sequence of a pumping asset and the start of the starting sequence of another pumping asset.
DrainingTime	TIME	Time for which Drain Valve is open when stopping the pumping asset.
DelayAfterValveOpen	TIME	Delay time before next action after opening valves.
DelayAfterValveClose	TIME	Delay time before next action after closing valves.
DelayAfterPumpStart	TIME	Delay time before next action after starting pumps.
DelayAfterPumpStop	TIME	Delay time before next action after stopping pumps.
EnabledStrategies	INT	Determines strategies to be enabled. • 1 = Priority only • 2 = Balance only • 3 = Operation Count only • 4 = Idle Time only • 5 = SwitchOver Any other value all = All enabled
OutValveFirst	BOOL	1 = Outlet valve is opened before starting the pump and closed after stopping the pump.

Outputs

Output Parameter Description

Parameter	Type	Description
ConfigError	INT	Configuration *error* detection code.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
IOPar	PumpSetGP_IOPar_DDT	Input and output parameters
PumpSetData	PumpSetDataGP_DDT	Data transfer structure between PUMPSETPATTERNGP DFB and PUMPINGASSET05GP DFB.
EMSC	EMCTLGP_SC_DDT	Equipment module sequence control structure, page 859.
EMST	EMCTLGP_ST_DDT	Provides the necessary data for monitoring and controlling Pump Set Equipment module, page 856.
EMCFG	EMCTLGP_CFG_DDT	Provides the necessary data for background monitoring, page 859.

PumpSetGP_IOPar_DDT Type

Name	Bit Name	Bit	Type	Description
PumpsetOperation-Time	—	—	ARRAY [0...14] OF TIME	Cumulative operation time for pumps as output parameter.
	PumpOpera-tionTime [{X}]	—	TIME	Pump {Y} operation time
	PumpOpera-tionTime [{Z}]	—	TIME	Unused
AssetDIAGState	—	—	ARRAY [0...14] OF INT	Pumping asset diagnosis state as output parameter.
	AssetDIAG-State[{X}]	—	INT	Pumping asset {Y} diagnosis state
	AssetDIAG-State[{Z}]	—	INT	Unused
PumpPriority	—	—	ARRAY [0...14] OF INT	Pumping asset priority as input parameter. The minimum value of the array elements is 1.
	PumpPriori-ty[{X}]	—	INT	Pumping asset {Y} priority
	PumpPriori-ty[{Z}]	—	INT	Unused
ActivePumpsSP	—	—	INT	Number of active pumps set point as input parameter. Minimum value of the ActivePumpsSP is 1 and the maximum value is the number of configured pumps.
RequestedPumps	—	—	INT	Number of currently requested pumps to be run as output parameter.
ActivePumps	—	—	INT	Number of pumps currently active as output parameter.
EMDiagState	—	—	INT	PumpSet equipment module diagnosis state as output parameter.
ResetOperation-Time	—	—	BOOL	Reset cumulative pumps operating time as input parameter.
PumpOperation-Count	—	—	ARRAY [0...14] OF DINT	Operation count of pumps as output parameter.
	PumpPriori-ty[{X}]	—	DINT	Pump {Y} operation count
	PumpPriori-ty[{Z}]	—	DINT	Unused
PumpIdleTime	—	—	ARRAY [0...14] OF TIME	Idle time of pumps as output parameter.
	PumpIdle-Time[{X}]	—	TIME	Pump {Y} idle time
	PumpIdle-Time[{Z}]	—	TIME	Unused
PumpRunTime	—	—	ARRAY [0...14] OF TIME	Operation time of pumps as output parameter.
	PumpRunTime [{P}]	—	TIME	—
ResetOperation-Count	—	—	BOOL	Reset pumps operation count as input parameter.
ResetIdleTime	—	—	BOOL	Reset pumps idle time as input parameter.
ResetRunTime	—	—	BOOL	Reset pumps operation time as input parameter.

Name	Bit Name	Bit	Type	Description
SwitchOverTime	—	—	TIME	Pumpset switch over time as input parameter.
NOTE: • X ranges from 0 to 9. • Y ranges from 1 to 10. • Z ranges from 10 to 14. • P ranges from 0 to 14.				

PumpSetDataGP_DDT Type

Name	Bit Name	Bit	Type	Description
GenericData	—	—	ARRAY [0...35] OF INT	Common data to be transferred between PUMPSETPATTERNGP DFB and PUMPINGASSET05GP DFB.
	GenericData [0]	—	INT	Bit 0 for pumpset pattern initialization
	GenericData [1]	—	INT	Group number
	GenericData [2]	—	INT	EM state
	GenericData [3]	—	INT	• 1 = Running • 2 = Holding • 3 = Restarting • 4 = Aborting • 5 = Stopping
	GenericData [4]	—	INT	Unused
	GenericData [5]	—	INT	• Bit 0 : Reset Operation Time • Bit 1 : Reset Operation Count • Bit 2 : Reset Idle Time • Bit 3 : Reset Operation Time from Program • Bit 4 : Reset Operation Count from Program • Bit 5 : Reset Idle Time from Program • Bit 6 : Reset Run Time • Bit 7 : Reset Run Time from Program
	GenericData [6]	—	INT	Unused
	GenericData [7]	—	INT	Unused
	GenericData [8]	—	INT	Unused
	GenericData [9]	—	INT	Unused
	GenericData [10]	—	INT	Full Active Assets of Group 1
	GenericData [11]	—	INT	Full Active Assets for EM of Group 1
	GenericData [12]	—	INT	Seq Active Assets of Group 1
	GenericData [13]	—	INT	Idle pumps of Group 1
	GenericData [14]	—	INT	Configuration *error* detection of Group 1
	GenericData [15]	—	INT	Configured pumps of Group 1

Name	Bit Name	Bit	Type	Description
	GenericData [16]	—	INT	Unused
	GenericData [17]	—	INT	Unused
	GenericData [18]	—	INT	Unused
	GenericData [19]	—	INT	Unused
	GenericData [20]	—	INT	Full Active Assets of Group 2
	GenericData [21]	—	INT	Full Active Assets for EM of Group 2
	GenericData [22]	—	INT	Seq Active Assets of Group 2
	GenericData [23]	—	INT	Idle pumps of Group 2
	GenericData [24]	—	INT	Configuration *error* detection of Group 2
	GenericData [25]	—	INT	Configured pumps of Group 2
	GenericData [26]	—	INT	Unused
	GenericData [27]	—	INT	Unused
	GenericData [28]	—	INT	Unused
	GenericData [29]	—	INT	Unused
	GenericData [30]	—	INT	Full Active Assets of Group 3
	GenericData [31]	—	INT	Full Active Assets for EM of Group 3
	GenericData [32]	—	INT	Seq Active Assets of Group 3
	GenericData [33]	—	INT	Idle pumps of Group 3
	GenericData [34]	—	INT	Configuration *error* detection of Group 3
	GenericData [35]	—	INT	Configured pumps of Group 3
PumpData	—	—	ARRAY [0...14,0...14] OF WORD	Pumping Asset Data
	PumpData [{X}]	—	ARRAY [0...14] OF WORD	Pumping asset {Y} data
		PumpData [{X}][0]	WORD	Pumping asset data
		PumpData [{X}][1]	WORD	• Bit 0 : Pump Remote • Bit 1 : Inlet Valve Remote • Bit 2 : Outlet Valve Remote • Bit 3 : Drain Valve Remote
		PumpData [{X}][2]	WORD	Unused
		PumpData [{X}][3]	WORD	Unused
		PumpData [{X}][4]	WORD	Unused

Name	Bit Name	Bit	Type	Description
		PumpData [{X}][5]	WORD	Unused
		PumpData [{X}][6]	WORD	Unused
		PumpData [{X}][7]	WORD	Run time high word
		PumpData [{X}][8]	WORD	Run time low word
		PumpData [{X}][9]	WORD	Operation on time high word
		PumpData [{X}][10]	WORD	Operation on time low word
		PumpData [{X}][11]	WORD	Operation count high word
		PumpData [{X}][12]	WORD	Operation count low word
		PumpData [{X}][13]	WORD	Idle time high word
		PumpData [{X}][14]	WORD	Idle time low word
EngParam	—	—	PumpSetGP_ EngPar_DDT	Pump set engineering parameters.
	SwitchOnDe-lay	—	TIME	Delay between the end of the starting sequence of a pumping asset and the start of the starting sequence of another pumping asset.
	Draining-Time	—	TIME	Time for which Drain Valve is open when stopping the pumping asset.
	DelayAfter-ValveOpen	—	TIME	Delay time before next action after opening valves.
	DelayAfter-ValveClose	—	TIME	Delay time before next action after closing valves.
	DelayAfter-PumpStart	—	TIME	Delay time before next action after starting pumps.
	DelayAfter-PumpStop	—	TIME	Delay time before next action after stopping pumps.
	Enable-dStrategies	—	INT	Determines strategies to be enabled. • 1 = Priority only • 2 = Balance only • 3 = Operation Count only • 4 = Idle Time only • 5 = Switch only • Any other value all = All enabled
	OutValve-First	—	BOOL	1 = Outlet valve is opened before starting the pump and closed after stopping the pump.

NOTE:
- X ranges from 0 to 14.
- Y ranges from 1 to 15.

Control — PUMPINGASSET05GP

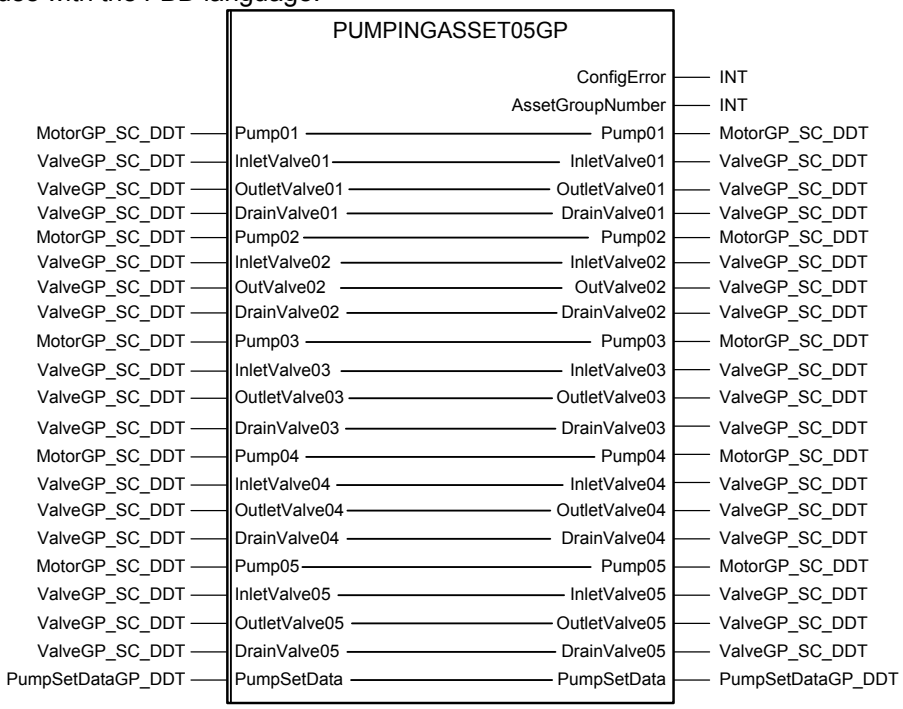
Overview

This chapter describes the PUMPINGASSET05GP DFB.

DFB Representation

Representation

The DFB that is used in the program has the following aspect at the section level. You can use it in any of the programming languages, although it is designed for use with the FBD language.



Outputs

Output Parameter Description

Parameter	Type	Description
`ConfigError`	`INT`	Configuration *error* detection code.
`AssetGroupNumber`	`INT`	Group number assigned as per the initialization sequence.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
`Pump01`	`MotorGP_SC_DDT, page 148`	Asset 01 pump
`InletValve01`	`ValveGP_SC_DDT, page 239`	Asset 01 inlet valve
`OutletValve01`	`ValveGP_SC_DDT, page 239`	Asset 01 outlet valve
`DrainValve01`	`ValveGP_SC_DDT, page 239`	Asset 01 drain valve
`Pump02`	`MotorGP_SC_DDT, page 148`	Asset 02 pump
`InletValve02`	`ValveGP_SC_DDT, page 239`	Asset 02 inlet valve

Parameter	Type	Description
OutletValve02	ValveGP_SC_DDT, page 239	Asset 02 outlet valve
DrainValve02	ValveGP_SC_DDT, page 239	Asset 02 drain valve
Pump03	MotorGP_SC_DDT, page 148	Asset 03 pump
InletValve03	ValveGP_SC_DDT, page 239	Asset 03 inlet valve
OutletValve03	ValveGP_SC_DDT, page 239	Asset 03 outlet valve
DrainValve03	ValveGP_SC_DDT, page 239	Asset 03 drain valve
Pump04	MotorGP_SC_DDT, page 148	Asset 04 pump
InletValve04	ValveGP_SC_DDT, page 239	Asset 04 inlet valve
OutletValve04	ValveGP_SC_DDT, page 239	Asset 04 outlet valve
DrainValve04	ValveGP_SC_DDT, page 239	Asset 04 drain valve
Pump05	MotorGP_SC_DDT, page 148	Asset 05 pump
InletValve05	ValveGP_SC_DDT, page 239	Asset 05 inlet valve
OutletValve05	ValveGP_SC_DDT, page 239	Asset 05 outlet valve
DrainValve05	ValveGP_SC_DDT, page 239	Asset 05 drain valve
PumpSetData	PumpSetData_GP_DDT	

NOTE: For additional information of those parameters that belong to ValveGP_SC_DDT, page 239 type.

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the pump set pattern.

Genies

Genie Properties

Refer to Genie Properties, page 62.

Genie Icons

Refer to Genie Icons, page 63.

Representation

Refer the genies of gpl_eqipmentmodule, page 862

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

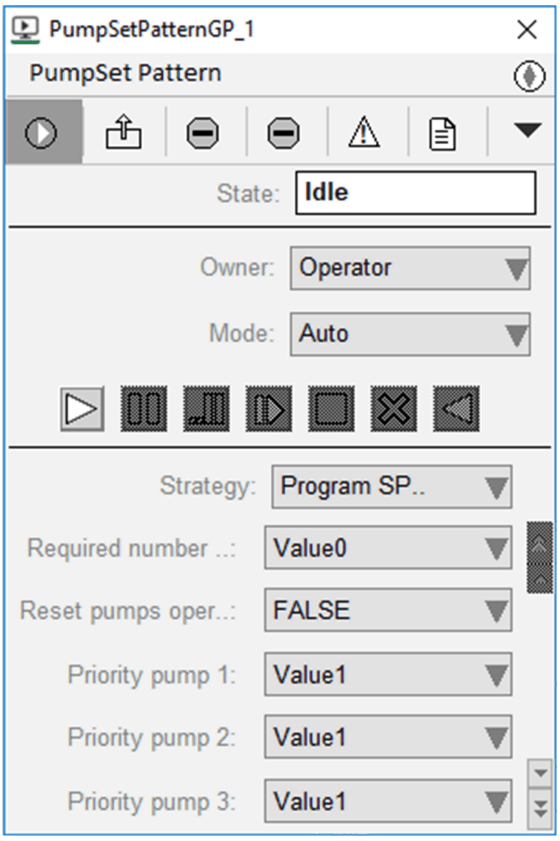
Available Tabs

During operation, clicking a pump set pattern genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Output Parameter
  - State Machine
  - Alarm, page 67
  - Event, page 69
- Tabs for optional functions, which appear only if configured:
  - InitCond, page 867
  - Failure, page 78

Operator Tab

The figure shows an operator tab:

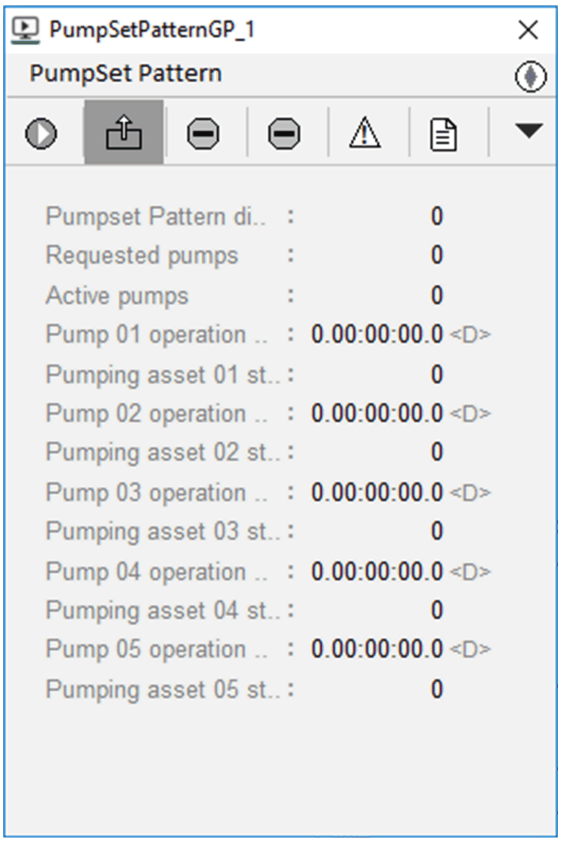


The table describes the control function of an operator tab:

Lable	Item Name	Description	Security parameter, page 787
State	State	State	–
Owner	OwnerSelect	Current owner of flow control pattern. For more details, refer to operating owner, page 58.	Set owner
Mode	Mode	Allows you to view and change the mode	Set auto mode, Set semi mode, set manual
Strategy	Strategy, OwnerSelect	Strategy of equipment module	Set strategy

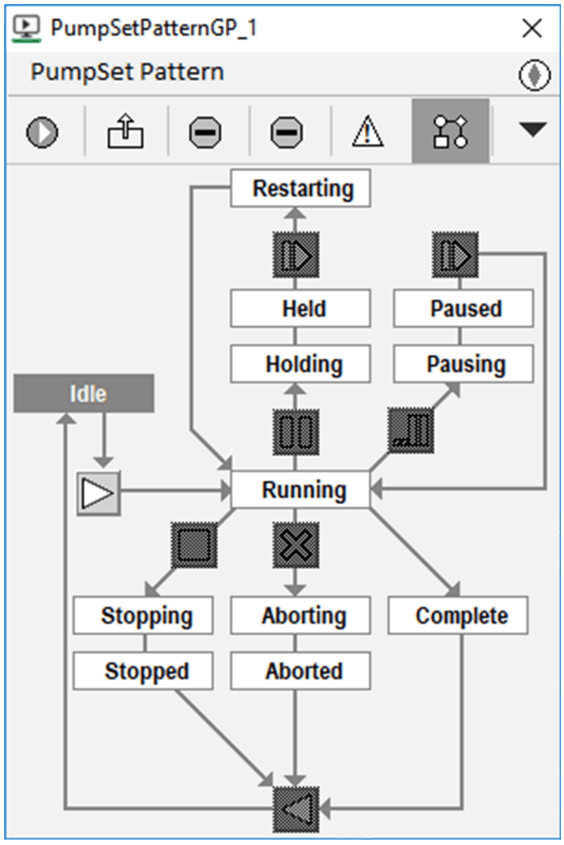
Output ParameterTab

The figure shows an output parameter tab:



State Machine Tab

The figure shows a state machine tab:



Items

Overview

This section describes the variables, equipstate variable, alarms, and events items of `pump set pattern`.

Variables

The table describes the variable items that are used by supervision components:

Item Name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`Transition{x}`	{Y} transition status	Done/ Pending	*{z}*
NOTE: {x} represents from 01 to 06. {y} represents from first to sixth. {z} represents from 0 to 5.			
`AutoResetDisable`	Auto reset disable status	Disabled/ Not Disabled	*Bit15*

Item Name	Description	Enumeration	Address
Bit items derived from ConfigurationWord.			
OwnerSelect	Owner of control module	Operator/ Program	*Bit0*
SemiAutomatic	Semi automatic mode command	SemiAutomatic/ Not Semi Automatic	*Bit1*
Manual	Equipment module sequence manual mode command	Auto/ Manual	*Bit2*
NextStep	Equipment module sequence next step command	On/ Off	*Bit3*
ScrollUp	Equipment module sequence scroll up command		*Bit4*
ScrollDown	Equipment module sequence scroll down command		*Bit5*
IO Items			
StatusWord	Status word	–	EMCTLGP_ST.STW
Configuration-Word	Configuration word		EMCTLGP_ST.CFGW
Calculated items: Refer to calculated items of equipment module control, page 867			

Alarms

The table describes the advanced alarm items that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the state:

Item name	Description	Expressions
HeldForError	Held for error	GPL_DualBitCheck (State, 3,9,1)

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
SateAlarm	All changes in the equipment module pattern status and output.
ModeAlarm	All changes in the mode and owner.
ActionAlarm	All actions by the operator.

Auxiliary Functions of Pump Set Equipment Module

What's in This Chapter

PUMPINGASSETSEQGP 804

Overview

This section describes the function blocks that supplement the PUMPSETPATTERN DFB and that have the objective of facilitating the programming of the equipment module.

PUMPINGASSETSEQGP

Overview

This section describes the PUMPINGASSETSEQGP DFB.

Description

General

It is an unprotected function block, that is used to control the start and stop sequence of a pumping asset. It evaluates the alert condition of the pumping asset diagnosis state.

Pumping Asset Sequence Execution

The Operation sequence to start a pumping asset is as follows:

- If used, close the Drain valve, wait for the Drain Valve to get closed (plus the delay time given by DelayAfterValveClose) or a detected failure.
- If OutValveFirst engineering parameter is True:
  - If used, open the Inlet valve, wait for the inlet valve to open (plus the delay time for each given by DelayAfterValveOpen) or for a detected failure.
  - If used, open the Outlet valve, wait for the outlet valve to open (plus the delay time for each given by DelayAfterValveOpen) or for a detected failure.
  - Start Pump, wait for the Pump running (plus the delay time given by DelayAfterPumpStart) or for a detected failure.
- If OutValveFirst engineering parameter is False:
  - If used, open the Inlet Valve, wait for the Inlet Valve to open (plus the delay time given by DelayAfterValveOpen) or for a detected failure.
  - Start Pump, wait for the Pump running (plus the delay time given by DelayAfterPumpStart) or for a detected failure.
  - If used, open the Outlet Valve, wait for the Outlet Valve to open (plus the delay time given by DelayAfterValveOpen) or for a detected failure.
- Wait for the delay time in case if it is needed to start more pumps.

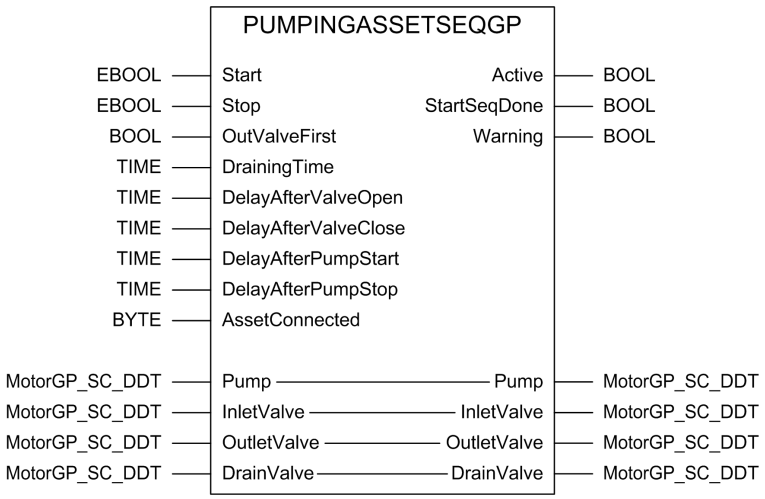
The Operation sequence to stop a pumping asset is as follows:

- If `OutValveFirst` engineering parameter is `True`:
  - Stop the pump, wait for the pump to stop (plus the delay time given by `DelayAfterPumpStop`) or for a detected failure.
  - If used, close the Inlet valve, wait for the inlet valve to close (plus the delay time for each given by `DelayAfterValveClose`) or for a detected failure.
  - If used, close the Outlet valve, and wait for the outlet valve to close (plus the delay time for each given by DelayAfterValveClose) or for a detected failure.
  - If used, open the Drain and wait for the Drain Valve to open (plus the delay time given by `DrainingTime`) or for a detected failure. Wait for the configured `DrainingTime` and then close the Drain Valve, wait for the Drain Valve to close (plus the delay time given by `DelayAfterValveClose`) or for a detected failure.
- Else, if `OutValveFirst` engineering parameter is `False`:
  - If used, close the Outlet Valve, wait for the Outlet Valve to get closed (plus the delay time given by `DelayAfterValveClose`) or a detected failure.
  - Stop the Pump, and wait for the Pump to stop (plus the delay time given by `DelayAfterPumpStop`) or for a detected failure.
  - If used, open the Drain valve, and wait for the Drain Valve to open (plus the delay time given by `DrainingTime`) or a detected failure. Wait for the configured `DrainingTime` and then close the Drain Valve, wait for the Drain Valve to close (plus the delay time given by `DelayAfterValveClose`) or for a detected failure.
  - If used, close the Inlet Valve, wait for the Inlet Valve to close (plus the delay time given by `DelayAfterValveClose`) or for a detected failure.

DFB Representation

Representation

The DFB that is used in the program has the following aspect at the section level. You can use it in any of the programming languages, although it is designed for use with the FBD language.



Inputs

Input Parameter Description

Parameter	Type	Description
Start	EBOOL	Trigger the start operation.
Stop	EBOOL	Trigger the stop operation.
OutValveFirst	BOOL	1 = Outlet valve is opened before starting the pump and closed after stopping the pump.
DrainingTime	TIME	Time for which drain valve will be open when stopping the pumping asset.
DelayAfterValveOpen	TIME	Delay time before next action after opening valves.
DelayAfterValveClose	TIME	Delay time before next action after closing valves.
DelayAfterPumpStart	TIME	Delay time before next action after starting pumps.
DelayAfterPumpStop	TIME	Delay time before next action after stopping pumps.
AssetConnected	BYTE	Indicates whether the components of a pumping asset are connected or not.

Outputs

Output Parameter Description

Parameter	Type	Description
Active	BOOL	Indicates the status of the pumping asset sequence: • True when start sequence is started. • False only when stop sequence is fully executed.
StartSeqDone	BOOL	Indicates that the start sequence has executed completely.
Warning	BOOL	Alerts based on each equipment availability/ detected failure/ alarm / owner / interlock.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
Pump	MotorGP_SC_DDT	Pump
InletValve	MotorGP_SC_DDT	Inlet valve
OutletValve	MotorGP_SC_DDT	Outlet valve
DrainValve	MotorGP_SC_DDT	Drain valve

NOTE: For additional information of those parameters that belong to MotorGP_SC_DDT type, page 148.

Flow Control Equipment Module

What's in This Chapter

Description of Flow Control Equipment Module 807
Strategy Selection and Functionality 808
Disabling the Strategies Based on Equipment Module State 809
Input and Output Parameters 809
Input Parameters Applicable For Each Strategy 812
Flow Control Equipment Module State Machine 812
Setpoint Management 812
Flow Control Equipment Module Diagnosis State 814
Flow Control Pumping Asset Diagnosis State 814
Modifying the `FlowCtlPattern` Template 815

Overview

This chapter describes the Flow Control Equipment Module management.

Description of Flow Control Equipment Module

Overview

It is an unprotected function block which is derived from `EMLOGIC` user-defined DFB. The `FLOWCTLPATTERN` DFB checks the status and controls the operation of control modules (Motors/Valves/Sensors, etc.) and services (Initial Conditions/ Detected Failure Conditions, etc.) to form a Flow Control Equipment Module.

A variable speed pump, an inlet valve, an outlet valve, and a drain valve are collectively referred to as a pumping asset. To configure a pumping asset, connecting a pump is mandatory whereas the valves are optional.

The `FLOWCTLPATTERN` DFB along with `FLOWASSET05GP` DFB can be used to run common pumping applications. The maximum number of pumping assets supported for operation is 10. The number of pumping assets can be modified according to the users application requirement.

Function Description

The main functions of the DFB are described in the following table:

Function	Description
Incorrect Configuration	Determines the total number of pumps connected to the `FlowAsset05GP` DFB from the connections to the pump input pins If the pump connections are not in order, it detects an incorrect configuration. For example, If `PUMP01`, `PUMP02`, `PUMP04` input pins are connected, the incorrect configuration is detected since the pumps are not connected sequentially. When there is a incorrect configuration is detected, Flow Control equipment module is not available for operation.
Order Sequence	The order sequence arranges the pumping assets in the order of minimum operation time/ highest priority/ minimum operation count / maximum idle time after operation first to maximum operation time/ least priority/ maximum operation count / minimum idle time based on the type of strategy run. In case of priority type, lower priority value corresponds to higher priority. If a balance strategy is selected, the order sequence is arranged by operation time. If a operation count strategy is selected the order sequence is arranged by operation count. If a idle time strategy is selected the order sequence is arranged by idle time after operation. If there are multiple pumping assets with same operation time or operation count or idle time, then the first connected pumping asset is considered to have a higher preference based on the strategy selected. If a priority strategy is selected, the order sequence is arranged by priority. If there are multiple pumping assets with the same priority, then for each set of same priority pumping assets, lesser operation time is

Function	Description
	considered as higher priority. Furthermore, in a set of pumping assets with same priority, if there are multiple pumping assets with same operation time, then the first connected pumping asset is considered to have a higher priority.
Switchover	If an undesirable condition is detected in an active pumping asset in the field, a Stop command is issued from the Flow Control Equipment Module to the inoperable pumping asset and the next available pumping asset in the order sequence is started. When the detected fault status of a pumping asset is reset, it automatically goes to Idle mode and waits for the command from the next calculation/ switching caused by the following factors: • Change in flow setpoint requires a change in number of pumps. • Detected failure of an active pump and switch over to an Idle one. • Recalculation is requested or forced. **NOTE:** Switchover does not apply if a user has paused/stopped/held a strategy during its execution.
Operation	The Flow Control Equipment Module can be run in Operator or Program owner. Strategies are used to run and stop the pumping asset, and once it achieves the target (or target not achieved), the Strategy is completed and the equipment module state reflects the reality of the field. • Incorrect configuration or detected failure condition will force the Flow Control Equipment Module to Stopped state. • If the Flow Control Equipment Module is in Stopped state, and the initial conditions are not satisfied (`INITCONDFAIL` is TRUE), then the Flow Control Equipment Module will go to Off state, and no command is possible, until the initial conditions are satisfied. • When Flow Control Equipment Module is in `STOPPED`/ `OFF` state, if any pumping asset is running outside the control of Equipment Module, then Flow Control Equipment Module will be in Running Externally state. • In Stopped state, the Flow Control Equipment Module can be started by selecting a strategy. Once a strategy is completed, if the Flow Control Equipment Module is not able to achieve the desired flow based on the required number of pumps, then it will generate a detected failure and then it will take corrective action when a pumping asset becomes available for it to control. • Recalculation Strategy will capture the Priority/ operation time/operation count / idle time of the pumping assets, refresh the order sequence and evaluate the assets which need to run and those that can be stopped. • Whenever a change in flow setpoint requires a change in number of pumps, recalculation will be executed. • The pumping assets are started as per the order sequence. • The execution of Stop Strategy, or Abort command during a strategy execution, or detected failure condition will stop the pumping assets simultaneously. • Pumps running outside the control of Flow Control Equipment Module are not considered for setpoint management.
Recalculation	Recalculation of the order sequence can be triggered either by the execution of the Recalculate strategy, or by the rising edge of the input pin `RECALCULATE` or change in the flow set point. If the equipment module state was of Program type prior to recalculation, then based on the change of Flow Set Point, required number of pumps changes thus leads to execution of Recalculation strategy. After recalculating the order sequence, it will try to achieve the set point (by starting the higher preference assets or stopping the lower preference assets). Once set point is achieved, it evaluates whether switch over is possible, by checking if an asset with higher preference in the order sequence is available and is idle, and pumping asset with lesser preference is available and active.
Detected Failure Condition	If a detected failure occurs then the Flow Control Equipment Module follows the Stop strategy execution matrix (Stopping to Stopped).

NOTE: For all the `MOTORGP` and `MOTORVSGP` instances connected to the Flow Control, the `FAILREARMEN`, `CONFREARMEN`, and `FAILEN` pins should be `TRUE`. Also, the `TIMEOUT` value should be greater than 0 seconds.

NOTE: The Flow Control Equipment Module checks for the interlocks. If any asset is interlocked while in operation, the next available asset is taken into operation and on restoration of the interlock condition or on bypass ,the pumping asset starts again if the pumping asset has highest rank in the order sequence.

Strategy Selection and Functionality

Description

The given below table explains the selection Strategy and its functionality.

Strategy	Strategy Number	Action
Program SP/Balance	1	The required number of pumps is calculated from the value of input pin `FLOWSP`. This strategy is only enabled in case that the `FLOWSP` input pin has been connected. In this strategy, the order sequence is determined by pump operation time.
Operator SP/Balance	2	The required number of pumps are calculated from the value of `IOPAR.FLOWSP` input parameter. In this strategy, the order sequence is determined by pump operation time.
Program SP/Priority	3	The required number of pumps are calculated from the value of input pin `FLOWSP`. This strategy is only enabled in case that the `FLOWSP` input pin has been connected. In this strategy, the order sequence is determined by pump priority.
Operator SP/Priority	4	The required number of pumps are calculated from the value of `IOPAR.FLOWSP` input parameter. In this strategy, the order sequence is determined by pump priority.
Program SP/OperCount	5	The number of active pumps set point is defined by input pin `FLOWSP`. This strategy is only enabled in case that the `FLOWSP` input pin has been connected. In this strategy, the order sequence is determined by pump operation count.
Operator SP/OperCount	6	The number of active pumps is defined by the `IOPAR.FLOWSP` Input Parameter.In this strategy, the order sequence is determined by pump operation count.
Program SP/Idletime	7	The number of active pumps set point is defined by input pin `FLOWSP`. This strategy is only enabled in case that the `FLOWSP` input pin has been connected. In this strategy, the order sequence is determined by pump idle time after operation.
Operator SP/Idletime	8	The number of active pumps is defined by the `IOPAR.FLOWSP` Input Parameter.In this strategy, the order sequence is determined by pump idle time after operation.
Program SP/SwitchOver	9	The number of active pumps set point is defined by input pin `FLOWSP`. This strategy is only enabled in case that the `FLOWSP` input pin has been connected. In this strategy, the pumping assets switches over as per the switch over time defined by the user. **NOTE:** The recalculate strategy works only for the change of switchover time in the strategy and no switchover of pumpset expected.
Operator SP/SwitchOver	10	The number of active pumps is defined by the `IOPAR.FLOWSP` Input Parameter. In this strategy, the pumping assets switches over as per the switch over time defined by the user. **NOTE:** The recalculate strategy works only for the change of switchover time in the strategy and no switchover of pumpset expected.
Recalculate	11	Forces the `FLOWCTLPATTERN` DFB to refresh the order sequence and perform the recalculation function.
Stop Equipment	12	Stop operation of all the pumping assets simultaneously.

Disabling the Strategies Based on Equipment Module State

Description

The disable strategies for Flow Control Equipment Module is same as Pump Set Pattern Equipment Module. For additional information, page 767.

Input and Output Parameters

Description

The Input parameters are as shown below:

Variable Name	Parameter Number	Data Type	Description
IOPAR.FLOWSP	1	REAL	Flow setpoint
IOPAR.RESETOPERATIONTIME	2	BOOL	Reset Pump Cumulative Run Time
IOPAR.RESETOPERATIONCOUNT	3	BOOL	Reset Pump Operation Count
IOPAR.RESETIDLETIME	4	BOOL	Reset Pump Idle Time
IOPAR.RESETRUNTIME	5	DURATION	Reset Pump Run time
IOPAR.SWITCHOVERTIME	6	DURATION	Switch over time
IOPAR.PUMPPRIORITY[0]	7	INT	Priority of Pumping Asset 01
IOPAR.PUMPPRIORITY[1]	8	INT	Priority of Pumping Asset 02
IOPAR.PUMPPRIORITY[2]	9	INT	Priority of Pumping Asset 03
IOPAR.PUMPPRIORITY[3]	10	INT	Priority of Pumping Asset 04
IOPAR.PUMPPRIORITY[4]	11	INT	Priority of Pumping Asset 05
IOPAR.PUMPPRIORITY[5]	12	INT	Priority of Pumping Asset 06
IOPAR.PUMPPRIORITY[6]	13	INT	Priority of Pumping Asset 07
IOPAR.PUMPPRIORITY[7]	14	INT	Priority of Pumping Asset 08
IOPAR.PUMPPRIORITY[8]	15	INT	Priority of Pumping Asset 09
IOPAR.PUMPPRIORITY[9]	16	INT	Priority of Pumping Asset 10

The Output parameters are as shown below:

Variable Name	Parameter Number	Data Type	Description
IOPAR.EMDIAGSTATE	1	ENUM	Pump Set Diagnosis State
IOPAR.FLOWPV	2	REAL	Flow process value
OPAR.CURRENTFLOWSP	3	REAL	Current flow setpoint
IOPAR.REQUIREDPUMPS	4	INT	Number of pumps required to meet the flow setpoint
IOPAR.ACTIVEFLOWPUMPS	5	INT	Number of active pumps to meet the flow setpoint
IOPAR.TOTALACTIVEPUMPS	6	INT	Total number of active pumps
IOPAR.PUMPSPEEDSP[0]	7	INT	Speed setpoint of the pump 01
IOPAR.PUMPOPERATIONTIME[0]/ IOPAR.PUMPOPERATIONCOUNT[0]/ IOPAR.PUMPIDLETIME[0]/IOPAR. PUMPRUNTIME[0]	8	DURATION /INT/ DURATION/ DURATION	Cumulative Run Time of Pump 01/ Operation Count for Pump 01 / Idle Time for Pump 01/ Run Time of Pump 01
IOPAR.ASSETDIAGSTATE[0]	9	INT	Diagnosis state of Pumping Asset 01 0: Normal, 1: Warning, 2: Failure
IOPAR.PUMPSPEEDSP[1]	10	REAL	Speed setpoint of the pump 02
IOPAR.PUMPOPERATIONTIME[1]/ IOPAR.PUMPOPERATIONCOUNT[1]/ IOPAR.PUMPIDLETIME[1]/IOPAR. PUMPRUNTIME[1]	11	DURATION /INT/ DURATION/ DURATION	Cumulative Run Time of Pump 02/ Operation Count for Pump 02 / Idle Time for Pump 02/ Run Time of Pump 02
IOPAR.ASSETDIAGSTATE[1]	12	INT	Diagnosis state of Pumping Asset 02 0: Normal, 1: Warning, 2: Failure
IOPAR.PUMPSPEEDSP[2]	13	REAL	Speed setpoint of the pump 03
IOPAR.PUMPOPERATIONTIME[2]/ IOPAR.PUMPOPERATIONCOUNT[2]/ IOPAR.PUMPIDLETIME[2]/IOPAR. PUMPRUNTIME[2]	14	DURATION /INT/ DURATION/ DURATION	Cumulative Run Time of Pump 03/ Operation Count for Pump 03 / Idle Time for Pump 03/ Run Time of Pump 03
IOPAR.ASSETDIAGSTATE[2]	15	INT	Diagnosis state of Pumping Asset 03 0: Normal, 1: Warning, 2: Failure

Variable Name	Parameter Number	Data Type	Description
IOPAR.PUMPSPEEDSP[3]	16	REAL	Speed setpoint of the pump 04
IOPAR.PUMPOPERATIONTIME[3]/ IOPAR.PUMPOPERATIONCOUNT[3]/ IOPAR.PUMPIDLETIME[3]/IOPAR. PUMPRUNTIME[3]	17	DURATION /INT/ DURATION/ DURATION	Cumulative Run Time of Pump 04/ Operation Count for Pump 04 / Idle Time for Pump 04/ Run Time of Pump 04
IOPAR.ASSETDIAGSTATE[3]	18	INT	Diagnosis state of Pumping Asset 04 0: Normal, 1: Warning, 2: Failure
IOPAR.PUMPSPEEDSP[4]	19	REAL	Speed setpoint of the pump 05
IOPAR.PUMPOPERATIONTIME[4]/ IOPAR.PUMPOPERATIONCOUNT[4]/ IOPAR.PUMPIDLETIME[4]/IOPAR. PUMPRUNTIME[4]	20	DURATION /INT/ DURATION/ DURATION	Cumulative Run Time of Pump 05/ Operation Count for Pump 05 / Idle Time for Pump 05/ Run Time of Pump 05
IOPAR.ASSETDIAGSTATE[4]	21	INT	Diagnosis state of Pumping Asset 05 0: Normal, 1: Warning, 2: Failure
IOPAR.PUMPSPEEDSP[5]	22	REAL	Speed setpoint of the pump 06
IOPAR.PUMPOPERATIONTIME[5]/ IOPAR.PUMPOPERATIONCOUNT[5]/ IOPAR.PUMPIDLETIME[5]/IOPAR. PUMPRUNTIME[5]	23	DURATION /INT/ DURATION/ DURATION	Cumulative Run Time of Pump 06/ Operation Count for Pump 06 / Idle Time for Pump 06/ Run Time of Pump 06
IOPAR.ASSETDIAGSTATE[5]	24	INT	Diagnosis state of Pumping Asset 06 0: Normal, 1: Warning, 2: Failure
IOPAR.PUMPSPEEDSP[6]	25	REAL	Speed setpoint of the pump 07
IOPAR.PUMPOPERATIONTIME[6]/ IOPAR.PUMPOPERATIONCOUNT[6]/ IOPAR.PUMPIDLETIME[6]/IOPAR. PUMPRUNTIME[6]	26	DURATION /INT/ DURATION/ DURATION	Cumulative Run Time of Pump 07/ Operation Count for Pump 07 / Idle Time for Pump 07/ Run Time of Pump 07
IOPAR.ASSETDIAGSTATE[6]	27	INT	Diagnosis state of Pumping Asset 07 0: Normal, 1: Warning, 2: Failure
IOPAR.PUMPSPEEDSP[7]	28	REAL	Speed setpoint of the pump 08.
IOPAR.PUMPOPERATIONTIME[7]/ IOPAR.PUMPOPERATIONCOUNT[7]/ IOPAR.PUMPIDLETIME[7]/IOPAR. PUMPRUNTIME[7]	29	DURATION /INT/ DURATION/ DURATION	Cumulative Run Time of Pump 08/ Operation Count for Pump 08 / Idle Time for Pump 08/ Run Time of Pump 08
IOPAR.ASSETDIAGSTATE[7]	30	INT	Diagnosis state of Pumping Asset 08 0: Normal, 1: Warning, 2: Failure
IOPAR.PUMPSPEEDSP[8]	31	REAL	Speed setpoint of the pump 09
IOPAR.PUMPOPERATIONTIME[8]/ IOPAR.PUMPOPERATIONCOUNT[8]/ IOPAR.PUMPIDLETIME[8]/IOPAR. PUMPRUNTIME[8]	32	DURATION /INT/ DURATION/ DURATION	Cumulative Run Time of Pump 09/ Operation Count for Pump 09 / Idle Time for Pump 09/ Run Time of Pump 09
IOPAR.ASSETDIAGSTATE[8]	33	INT	Diagnosis state of Pumping Asset 09 0: Normal, 1: Warning, 2: Failure
IOPAR.PUMPSPEEDSP[9]	34	REAL	Speed setpoint of the pump 10
IOPAR.PUMPOPERATIONTIME[9]/ IOPAR.PUMPOPERATIONCOUNT[9]/ IOPAR.PUMPIDLETIME[9]/IOPAR. PUMPRUNTIME[9]	35	DURATION /INT/ DURATION/ DURATION	Cumulative Run Time of Pump 10/ Operation Count for Pump 10 / Idle Time for Pump 10/ Run Time of Pump 10
IOPAR.ASSETDIAGSTATE[9]	36	INT	Diagnosis state of Pumping Asset 10: Normal, 1: Warning, 2: Failure

Input Parameters Applicable For Each Strategy

Description

The below table provides the information about the input parameters that are applicable for each strategy.

Strategy	Input Parameters Applicable
Program SP/Balance	2, 3, 4, 5
Operator SP/Balance	1, 2, 3, 4, 5
Program SP/Priority	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
Operator SP/Priority	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
Program SP/Operation Count	2, 3, 4, 5
Operator SP/Operation Count	1, 2, 3, 4, 5
Program SP/Idletime	2, 3, 4
Operator SP/Idletime	1, 2, 3, 4
Program SP/Idletime	2, 3, 4
Operator SP/Idletime	1, 2, 3, 4
Program SP/SwitchOver	2, 3, 4, 5, 6
Operator SP/SwitchOver	1, 2, 3, 4, 5, 6
Recalculate	2, 3, 4, 5, 6
Stop Equipment	2, 3, 4, 5, 6

Flow Control Equipment Module State Machine

Description

Flow Control follows the same Equipment Module State Machine as Pump Set Equipment Module. For additional information, page 770.

Setpoint Management

Description

Based on the flow setpoint value and the engineering parameters (MINIMUMSP, MAXIMUMSP, and OPTIMALSP), the Flow Control Equipment Module calculates the number of pumps and the speed required to achieve the desired flow setpoint.

The applicable range of values for the above mentioned items is given below:

- Flow setpoint: 0.0 - 100.0%
- Pump speed: 0.0 - 100.0%
- Maximum setpoint: 0.0 - 100.0%
- Minimum setpoint: 0.0 - Maximum setpoint value
- Optimal setpoint: Minimum setpoint to Maximum setpoint value

The calculation for number of pumps and their speed is governed by the following rules:

- Pump speed should never go under minimum setpoint, unless it is necessary to achieve flow setpoint.
- Pump speed should never go over maximum setpoint, unless it is necessary to achieve flow setpoint.
- Pumps running in control of flow control equipment module should always operate at same speed.
- Pump speed should go up to maximum setpoint before starting the next pump.
- Pump should stop when the remaining can work at optimal setpoint.

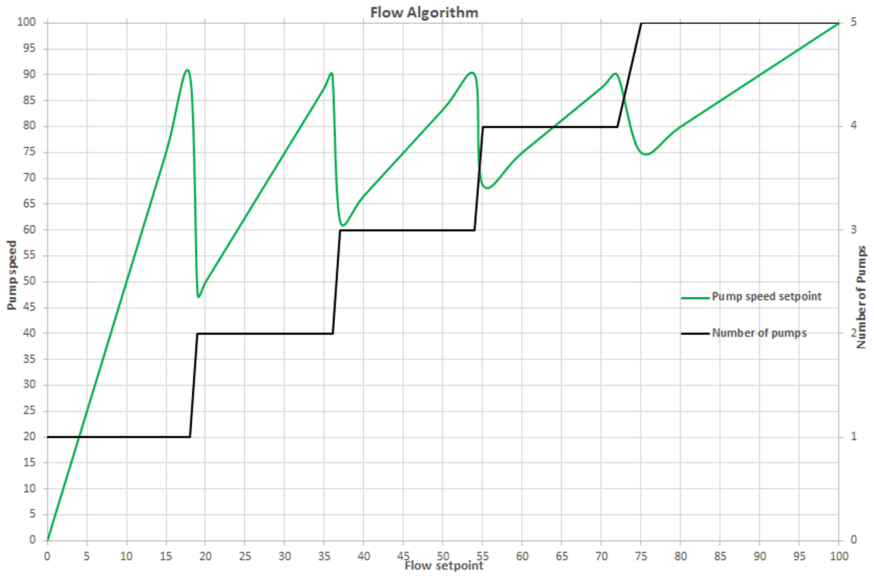
NOTE:

- This calculation takes place when the equipment module state is either running or configuring.
- If the optimal setpoint value is 0.0%, then pumps once started will not be stopped, since the remaining pumps cannot run at this optimal setpoint and achieve the flow setpoint.
- If the flow setpoint is 0.0%, then the calculation will result in one pump running at 0.0% speed.

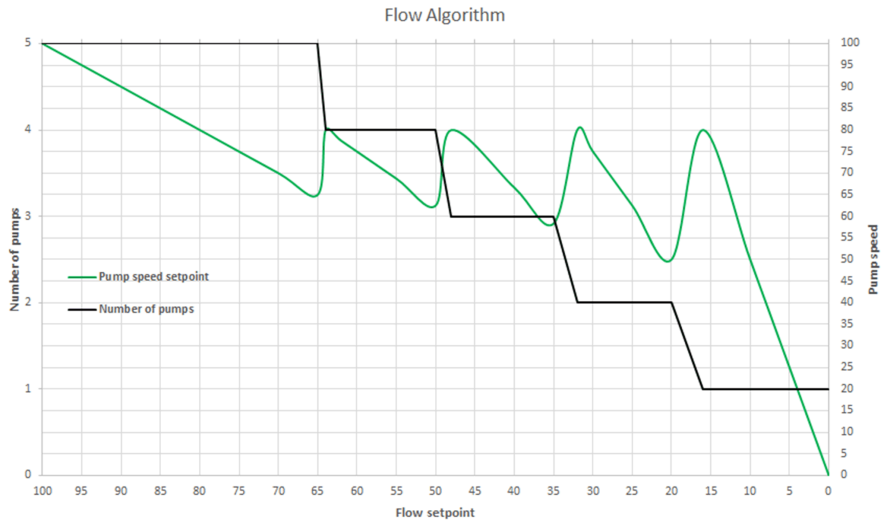
The below given graphs show the calculation result of increasing and decreasing flow setpoint. These graphs are based on the below configuration:

- Maximum setpoint – 90%
- Minimum setpoint – 40%
- Optimal setpoint – 80%
- Configured pumps – 5

Case 1: Increasing flow setpoint



Case 2: Decreasing flow setpoint



Flow Control Equipment Module Diagnosis State

Description

The Flow Control equipment module diagnosis state is evaluated as given below:

- `Normal`: State is valid if all below conditions are true:
  - When no pumping asset is running in external control mode or interlocked or remote Set point.
  - There are no detected failure and alert conditions.
- `Warning`: State is valid if any of below condition is true:
  - When total number of running pumping assets (inclusive of the pumps that are not in control of `FLOWCTLPATTERN`) are running more than number of assets requested by flow control pattern based on the current flow set point.
  - If stop strategy could not stop all the pumping assets.
  - If requested number of assets is equal to fully active assets for Equipment Module and there is a problem in any pumping asset.
- `Failure`: State is valid if any of below condition is true:
  - When there is an incorrect configuration.
  - When there is a detected failure condition.
  - When the number of active flow pumping asset is lesser than number of pumping assets required by flow control pattern to meet the flow setpoint.

Flow Control Pumping Asset Diagnosis State

Description

Each individual pumping asset has its own diagnosis state and each of them are evaluated as given below:

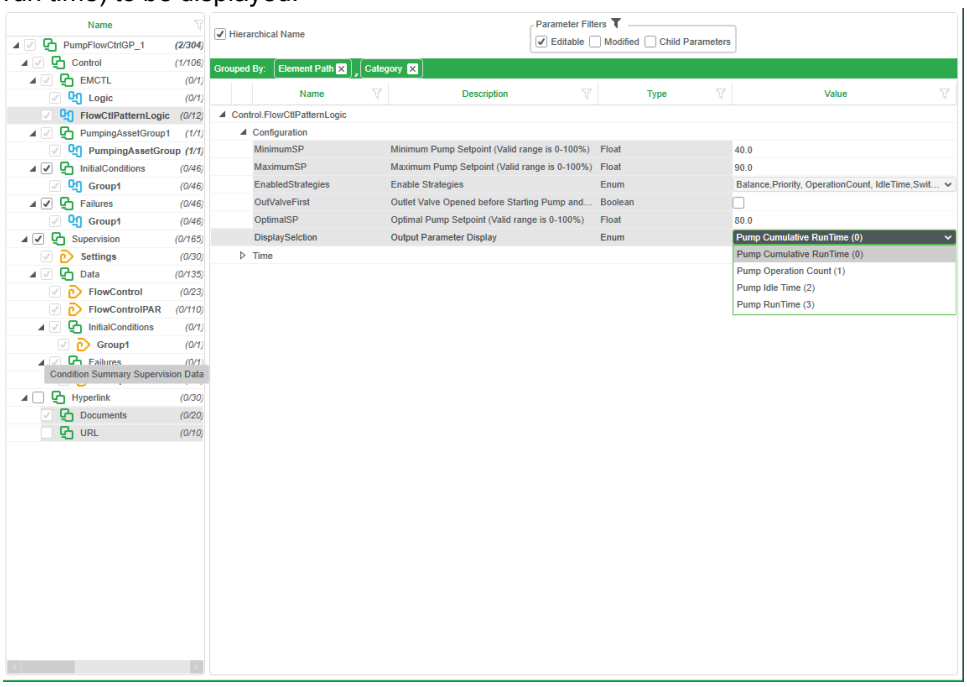
- `Normal`: The set of Valves and Pump is operating properly.

- `Warning`: The set of Valves and Pump caused a problem during its last operation. The last time that the algorithm tried to execute the complete sequence for starting/stopping the Pumping Asset caused a problem for some reason (for example, the operating mode was not correct, it triggered an alarm, the confirmations were not reached, etc...). Switching from alert to a detected failure does not reset the alert condition, the only way that the alert is reset is when the same Pumping Asset is operated again for starting or stopping it.
- `Failure`: Any one of the Valves or Pump is in inoperable condition (The `FAILD`, `ALARM`, `REARMR` or `OOS` value is TRUE).

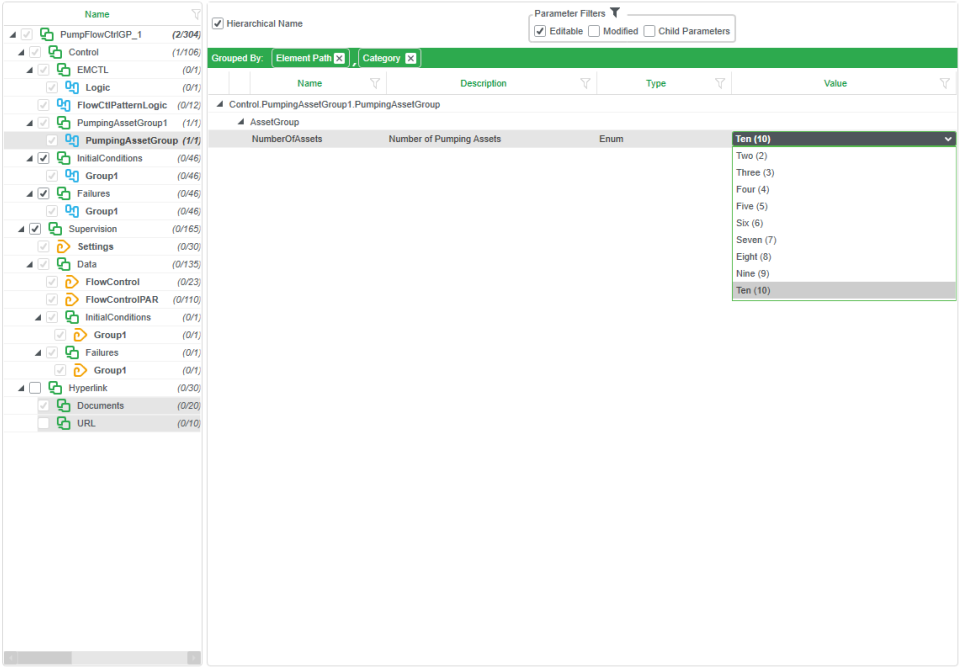
Modifying the `FlowCtlPattern` Template

Description

The flow control equipment module can handle up to 10 pumping assets and by default can be connected to 2 pumping assets also the user has the facility to select the required output parameter (operation time / operation count/ idle time/ run time) to be displayed.



The user can select the required number of pumping assets from 2 to 10 with help of parameter named Number of Pumping Assets.



Steps to modify `FlowCtlPattern` template for the Flow Control Operation

The `FlowCtlPattern` template is provided with 5 pumping assets. Each asset includes one variable speed pump and three valves (Inlet, Outlet and Drain valve). Steps to modify the `FlowCtlPattern` template are identical to the modification of `PumpSetPattern` template. For additionalinformation, page 772, refer to Modifying the PumpSetPattern Template section.

Perform the following steps to connect the 15 Pumping assets:

- Modify `$FlowAsset05GP_UL`.
- Update `$FlowAsset05GP_UC`.
- Add a new instance of `$FlowAsset05GP_UC` in `$FlowCtlPatternGP_UC`.
- Modify `$FlowCtlPatternPARGP_CD`.
- Update `$FlowCtlPatternGP_CD`.
- Update `$FlowCtlPatternGP_CS`.
- Update `$PumpFlowCtrlGP`.

Flow Control

What's in This Chapter

Template 817
Control — FLOWCTLPATTERNGP 823
Control — FLOWASSET05GP 829
Supervision 831

Overview

This chapter describes the components that provide the functionality of template, control services, and Supervision functions of the flow control.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$PumpFlowCtrlGP` template.

Description

General Description

The `$PumpFlowCtrlGP` template is used to monitor and manage the Flow Control operation. It is provided with the configuration of 5 pumping assets. Each asset includes one variable speed pump and three valves (Inlet, Outlet and Drain).

The composite and facet names without the **$** prefix can be modified, while the composite and facet names with **$** prefix are standard and cannot be modified.

Parameters

Parameters

The `$PumpFlowCtrlGP` template defines operation of a typical flow control pattern. Alternative flow control pattern facets enable the modification of the flow control pattern that are available.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Disable auto reset	`Boolean`	False	–
	Minimum pump setpoint (valid range is 0-100%)	`Float`	40.0	–
	Maximum pump setpoint (valid range is 0-100%)	`Float`	90.0	–
	Enable strategies	`Enum`	Priority and	–

Section	Parameter description	Type	Default	Additional remarks
			Balance (0)	
	Outlet valve opened before starting pump and closed after stopping pump	Boolean	False	–
	Optimal pump setpoint (Valid range is 0-100%)	Float	80.0	–
	Output parameter display	Enum	Pump operation count (1)	–
Time	Delay between 2 sequence of starting pumping asset	Duration	00:00:03	–
	Drain valve open time after stopping pumping asset		00:00:05	–
	Delay after opening valves		00:00:03	–
	Delay after closing valves		00:00:03	–
	Delay after starting pumps		00:00:03	–
	Delay after stopping pumps		00:00:03	–
AssetGroup	Number of PumpingAssets	Enum	Two (2)	–
Description	**InitialConditions**			
	Condition 1 to 15	String	–	–
	Group description		@(Initial condi-tions)	–
	FailureConditions			
	Condition 1 to 15	String	–	–
	Group description		@ (Fail-ures)	–
Reset Required	**InitialConditions**			
	Condition 1 to 15	Boolean	False	–
	FailureConditions			
	Condition 1 to 15	Boolean	False	–
Disable Bypass	**InitialConditions**			
	Condition 1 to 15	Boolean	False	–
	FailureConditions			
	Condition 1 to 15	Boolean	False	–

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Operation	Normal Owner	Enum	Program (1)	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	Set owner	Enum	Operator (1)	Privilege to change owner. This parameter is used in Operator tab, page 832.
	Set setpoint		Operator (1)	Privilege to set setpoint. This parameter is used in Operator tab, page 832.
	Set reset		Operator (Confirmed) (10)	Privilege to reset. This parameter is used in Operator tab, page 832.
	Set alarm acknowledge		Operator (1)	Privilege to acknowledge alarm. This parameter is used in Alarm tab, page 67.
	Set bypass conditions		Supervisor (2)	Privilege to bypass conditions.
	Set semi mode		Operator (1)	Privilege to change set semi mode. This parameter is used in Operator tab, page 832.
	Set manual mode			Privilege to change set manual mode. This parameter is used in Operator tab, page 832.
	Set next step			Privilege to change set next step. This parameter is used in Operator tab, page 832.
	Set scroll up			Privilege to change set scroll up. This parameter is used in Operator tab, page 832.
	Set scroll down			Privilege to change set scroll down. This parameter is used in Operator tab, page 832.
	Set start			Privilege to change set start. This parameter is used in Operator tab, page 832.
	Set hold			Privilege to change set hold. This parameter is used in Operator tab, page 832.
	Set restart			Privilege to change set restart. This parameter is used in Operator tab, page 832.
Security	Set stop	Enum	Operator (1)	Set reset
	Set reset		Operator (Confirmed) (10)	Privilege to reset. This parameter is used in Operator tab, page 832.
	Set pause		Operator (1)	Privilege to change set pause. This parameter is used in Operator tab, page 832.
	Set resume			Privilege to change set resume. This parameter is used in Operator tab, page 832.

Section	Parameter description	Type	Default	Additional remarks
	Set abort			Privilege to change set abort. This parameter is used in Operator tab, page 832.
	Set strategy			Privilege to change set strategy. This parameter is used in Operator tab, page 832.

Asset and Display Parameters

For more details, refer to the topic describing the parameters of asset and display, page 43.

Data

The table describes the data parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Disable start command button	`Boolean`	False	–
	Disable hold command button			–
	Disable pause command button			–
	Disable reset command button			–
	Disable stop command button			–
	Disable abort command button			–
	Disable restart command button			–
Alarm	Held for error severity	`Enum`	Low (4000)	–
	Held for error description		@(Held For Error)	–
	Held for error group	`Enum`	`Failure`	–
Event	Enable log	`Boolean`	False	Enable event logging for asset status.
Historize	State	`Boolean`	False	–
	Current step description			–
	Step description			–
	Next step description			–
	Transition description			–
	Disable strategy			–
	Equipment module state			–
	Equipment output status			–
	Sequence equipment module mode			–
	Equipment module strategy			–
	Command			–
	Elapsed time			–
Alarm	Equipment module disable	`Boolean`	True	–

Section	Parameter description	Type	Default	Additional remarks
	Equipment module discription	String	@(EM Alarm Description)	–
	Equipment module severity	Enum	Low (4000)	–
	Equipment module group	String	Failure	–
	Equipment module delay	Duration	00:00:00	–
	Asset 01 disable	Boolean	True	–
	...			–
	Asset 10 disable			–
	Asset 01 discription	String	@(Asset 01 Alarm Description)	–
	...		...	–
	Asset 10 discription		@(Asset 10 Alarm Description)	–
	Asset 01 severity	Enum	Low (4000)	–
	...			–
	Asset 10 severity			–
	Asset 01 group	String	Failure	–
	...			–
	Asset 10 group			–
	Asset 01 delay	Duration	00:00:00	–
	...			–
	Asset 10 delay			–
Alarm	Equipment module failure disable	Boolean	True	–
	Equipment module failure discription	String	@(EM Alarm Description)	–
	Equipment module failure severity	Enum	Low (4000)	–
	Equipment module failure group	String	Failure	–
	Equipment module failure delay	Duration	00:00:00	–
	Asset 01 failure disable	Boolean	True	–
	...			–
	Asset 10 failure disable			–
	Asset 01 failure discription	String	@(Asset 01 Failure Description)	–
	...		...	–
	Asset 10 failure discription		@(Asset 10 Failure Description)	–
	Asset 01 failure severity	Enum	Low (4000)	–
	...			–

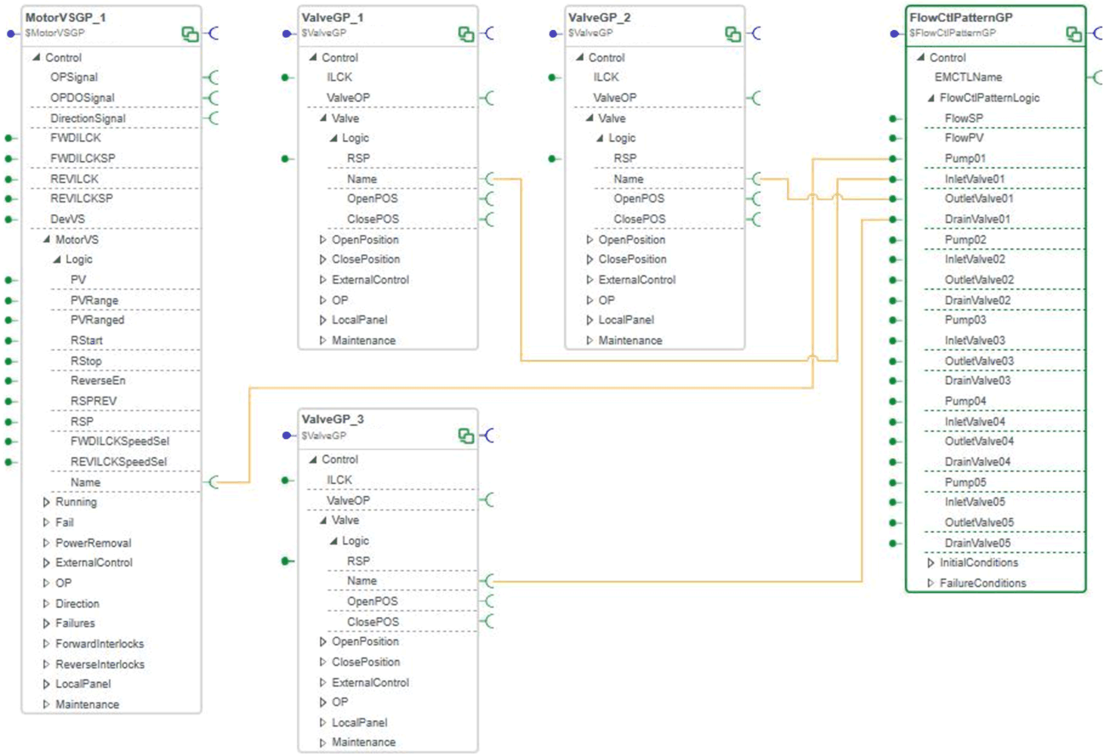
Section	Parameter description	Type	Default	Additional remarks
	`Asset 10 failure severity`			–
	`Asset 01 failure group`	`String`	`Failure`	–
	...			–
	`Asset 10 failure group`			–
	`Asset 01 failure delay`	`Dura-tion`	00:00:00	–
	...			–
	`Asset 10 failure delay`			–
Historize	**InitalConditionTags** and **Failure Conditions Tags**			
	Conditions	`Boolean`	False	–

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the `$PumpFlowCtrlGP` (also applicable for `$PumpFlowCtrlCE`) template as it appears in `Links Editor` and an example of other templates connected to it:



For interface identifiers, refer to the control module template interfaces, page 49.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control — FLOWCTLPATTERNGP

Overview

This chapter describes the FLOWCTLPATTERNGP DFB.

Description

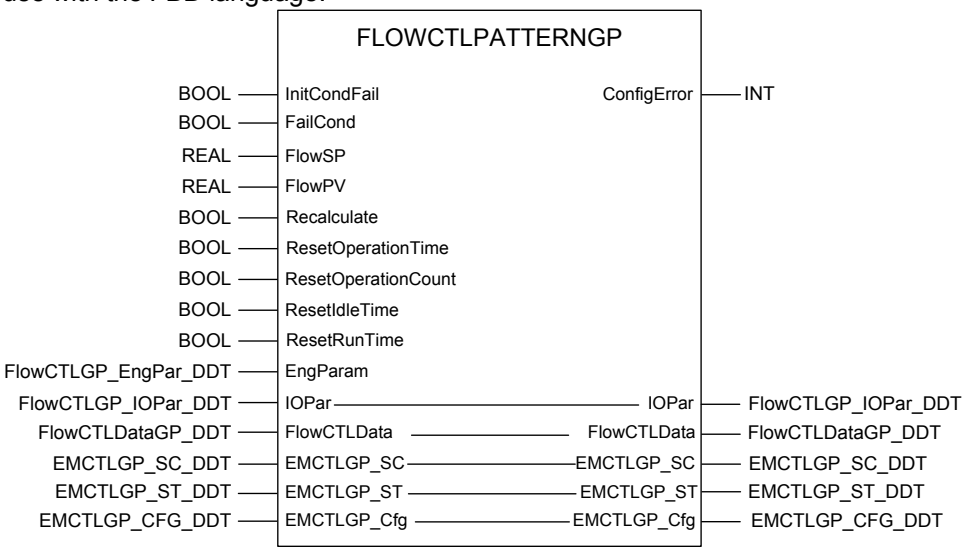
General

The FLOWCTLPATTERNGP DFB can be used to run common pumping applications like Sump management, Transfer and Circulation pump, etc. The maximum number of pumping assets supported for operation is five. The number of pumping assets can be modified according to the users application requirement.

DFB Representation

Representation

The DFB that is used in the program has the following aspect at the section level. You can use it in any of the programming languages, although it is designed for use with the FBD language.



Inputs

Input Parameter Description

Parameter	Type	Description
InitCondfail	BOOL	Detected failure in initial condition.
FailCond	BOOL	Detected failure condition.
FlowSP	REAL	Flow setpoint
FlowPV	REAL	Flow present value.
Recalculate	BOOL	Recalculate ON/OFF status of pumps based on current equipment module state.
ResetOperation-Time	BOOL	Reset the operation time of all the pumps.

Parameter	Type	Description
ResetOperation-Count	BOOL	Reset the operation count of all the pumps.
ResetIdleTime	BOOL	Reset the idle time of all the pumps.
ResetRunTime	BOOL	Reset the run time of all running pumps.
EngParam	FlowCTLGP_EngPar_DDT	Pump set engineering parameters. **NOTE:** If this input pin is not connected, then the following default values are considered by the DFB: • MinimuSP: 40.0% • OptimaSP: 80.0% • MaximumSP: 90.0% • SwitchOnDelay: 3s • DrainingTime: 5s • DelayAfterValveOpen: 3s • DelayAfterValveClose: 3s • DelayAfterPumpStart: 3s • DelayAfterPumpStop: 3s • EnabledStrategies: 0 • OutValveFirst: FALSE

FlowCTLGP_EngPar_DDT Type

Name	Type	Description
MinimumSP	REAL	Minimum pump setpoint
OptimalSP	REAL	Optimum pump setpoint
MaximumSP	REAL	Maximum pump setpoint
SwitchOnDelay	TIME	Delay between the end of the starting sequence of a pumping asset and the start of the starting sequence of another pumping asset.
DrainingTime	TIME	Time for which Drain Valve is open when stopping the pumping asset.
DelayAfterValveOp-en	TIME	Delay time before next action after opening valves.
DelayAfterValve-Close	TIME	Delay time before next action after closing valves.
DelayAfterPump-Start	TIME	Delay time before next action after starting pumps.
DelayAfterPumpStop	TIME	Delay time before next action after stopping pumps.
EnabledStrategies	INT	Determines strategies to be enabled. • 1 = Priority only • 2 = Balance only • 3 = Operation Count only • 4 = Idle Time only • 5 = SwitchOver • Any other value all = All enabled
OutValveFirst	BOOL	1 = Outlet valve is opened before starting the pump and closed after stopping the pump.

Outputs

Output Parameter Description

Parameter	Type	Description
ConfigError	INT	Configuration *error* detection code.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
IOPar	FlowCTLGP_IOPar_DDT	Input and output parameters
FlowCTLData	FlowCTLDataGP_DDT	Data transfer structure between FLOWCTLPATTERNGP DFB and FLOWASSET05GP DFB.
EMCTLGP_SC	EMCTLGP_SC_DDT	Equipment module sequence control structure.
EMCTLGP_ST	EMCTLGP_ST_DDT	Provides the necessary data for monitoring and controlling Pump Set Equipment module.
EMCTLGP_Cfg	EMCTLGP_CFG_DDT	Provides the necessary data for background monitoring.

FlowCTLGP_IOPar_DDT Type

Name	Bit Name	Bit	Type	Description
PumpsetOperation-Time	—	—	ARRAY [0...14] OF TIME	Cumulative operation time for pumps as output parameter.
	PumpOpera-tionTime [{X}]	—	TIME	Pump {Y} operation time
	PumpOpera-tionTime [{Z}]	—	TIME	Unused
PumpSpeedSP	—	—	ARRAY [0...14] OF REAL	Speed setpoint of the pumps as output parameters.
	PumpOpera-tionTime [{X}]	—	TIME	Pump {Y} speed setpoint
	PumpOpera-tionTime [{Z}]	—	TIME	Unused
PumpOperation-Count	—	—	ARRAY [0...14] OF DINT	Operation count of pumps as output parameter.
	PumpPriori-ty[{X}]	—	DINT	Pump {Y} operation count
	PumpPriori-ty[{Z}]	—	DINT	Unused
PumpIdleTime	—	—	ARRAY [0...14] OF TIME	Idle time of pumps as output parameter.
	PumpIdle-Time[{X}]	—	TIME	Pump {Y} idle time

Name	Bit Name	Bit	Type	Description
	PumpIdle-Time[{Z}]	—	TIME	Unused
PumpRunTime	—	—	ARRAY [0...14] OF TIME	Operation time of pumps as output parameter.
	PumpRunTime [{X}]	—	TIME	Pump {Y} run time
	PumpRunTime [{Z}]	—	TIME	Unused
FlowSP	—	—	REAL	Flow setpoint as input parameter.
CurrentFlowSP	—	—	REAL	Current flow setpoint as output parameter.
FlowPV	—	—	REAL	Flow process value as output parameter.
AssetDIAGState	—	—	ARRAY [0...14] OF INT	Pumping asset diagnosis state as output parameter.
	AssetDIAG-State[{X}]	—	INT	Pumping asset {Y} diagnosis state
	AssetDIAG-State[{Z}]	—	INT	Unused
PumpPriority	—	—	ARRAY [0...14] OF INT	Pumping asset priority as input parameter. The minimum value of the array elements is 1.
	PumpPriori-ty[{X}]	—	INT	Pumping asset {Y} priority
	PumpPriori-ty[{Z}]	—	INT	Unused
EMDiagState	—	—	INT	Flow control equipment module diagnosis state as output parameter.
RequiredPumps	—	—	INT	Flow control equipment module diagnosis state as output parameter
ActivFlowPumps	—	—	INT	Number of active pumps to meet the flow setpoint as output parameter.
TotalActivePumps	—	—	INT	Total number of active pumps as output parameter.
ResetOperation-Time	—	—	BOOL	Reset cumulative pumps operating time as input parameter.
ResetOperation-Count	—	—	BOOL	Reset pumps operation count as input parameter.
ResetIdleTime	—	—	BOOL	Reset pumps idle time as input parameter.
ResetRunTime	—	—	BOOL	Reset pumps operation time as input parameter.
SwitchOverTime	—	—	TIME	Pumpset switch over time as input parameter.

NOTE:

- X ranges from 0 to 9.
- Y ranges from 1 to 10.
- Z ranges from 10 to 14.

FlowCTLDataGP_DDT Type

Name	Bit Name	Bit	Type	Description
GenericData	—	—	ARRAY [0...40] OF INT	Common data to be transferred between FLOWCTLPATTERN DFB and FLOWASSET05GP DFB.
	GenericData [0]	—	INT	Bit 0 for pumpset pattern initialization
	GenericData [1]	—	INT	Group number

Name	Bit Name	Bit	Type	Description
	GenericData[2]	—	INT	EM state
	GenericData[3]	—	INT	• 1 = Running • 2 = Holding • 3 = Restarting • 4 = Aborting • 5 = Stopping
	GenericData[4]	—	INT	Unused
	GenericData[5]	—	INT	• Bit 0 : Reset Operation Time • Bit 1 : Reset Operation Count • Bit 2 : Reset Idle Time • Bit 3 : Reset Operation Time from Program • Bit 4 : Reset Operation Count from Program • Bit 5 : Reset Idle Time from Program • Bit 6 : Reset Run Time • Bit 7 : Reset Run Time from Program
	GenericData[6]	—	INT	Unused
	GenericData[7]	—	INT	Unused
	GenericData[8]	—	INT	Unused
	GenericData[9]	—	INT	Unused
	GenericData[10]	—	INT	Fully Active Assets of Group 1
	GenericData[11]	—	INT	Fully Active Assets for EM of Group 1
	GenericData[12]	—	INT	Seq Active Assets of Group 1
	GenericData[13]	—	INT	Idle pumps of Group 1
	GenericData[14]	—	INT	Configuration *error* detection of Group 1
	GenericData[15]	—	INT	Configured pumps of Group 1
	GenericData[16]	—	INT	Available groups of 1
	GenericData[17]	—	INT	SeqActiveAssetsforEM of group 1
	GenericData[18]	—	INT	Unused
	GenericData[19]	—	INT	Unused
	GenericData[20]	—	INT	Fully Active Assets of Group 2
	GenericData[21]	—	INT	Fully Active Assets for EM of Group 2
	GenericData[22]	—	INT	Seq Active Assets of Group 2
	GenericData[23]	—	INT	Idle pumps of Group 2
	GenericData[24]	—	INT	Configuration *error* detection of Group 2
	GenericData[25]	—	INT	Configured pumps of Group 2

Name	Bit Name	Bit	Type	Description
	GenericData [26]	—	INT	Available groups of 2
	GenericData [27]	—	INT	SeqActiveAssetsforEM of group 2
	GenericData [28]	—	INT	Unused
	GenericData [29]	—	INT	Unused
	GenericData [30]	—	INT	Fully Active Assets of Group 3
	GenericData [31]	—	INT	Fully Active Assets for EM of Group 3
	GenericData [32]	—	INT	Seq Active Assets of Group 3
	GenericData [33]	—	INT	Idle pumps of Group 3
	GenericData [34]	—	INT	Configuration *error* detection of Group 3
	GenericData [35]	—	INT	Configured pumps of Group 3
	GenericData [36]	—	INT	Available groups of 3
	GenericData [37]	—	INT	SeqActiveAssetsforEM of group 3
	GenericData [38]	—	INT	Unused
	GenericData [39]	—	INT	Unused
	GenericData [40]	—	INT	Unused
PumpData	—	—	ARRAY [0...14,0...14] OF WORD	Pumping Asset Data
	PumpData [{X}]	—	ARRAY [0...14] OF WORD	Pumping asset {Y} data
		PumpData [{X}][0]	WORD	Pumping asset data
		PumpData [{X}][1]	WORD	• Bit 0 : Pump Remote • Bit 1 : Inlet Valve Remote • Bit 2 : Outlet Valve Remote • Bit 3 : Drain Valve Remote
		PumpData [{X}][2]	WORD	Unused
		PumpData [{X}][3]	WORD	Unused
		PumpData [{X}][4]	WORD	Unused
		PumpData [{X}][5]	WORD	Pump speed setpoint high word
		PumpData [{X}][6]	WORD	Pump speed setpoint low word
		PumpData [{X}][7]	WORD	Run time high word
		PumpData [{X}][8]	WORD	Run time low word
		PumpData [{X}][9]	WORD	Operation on time high word

Name	Bit Name	Bit	Type	Description
		PumpData [{X}][10]	WORD	Operation on time low word
		PumpData [{X}][11]	WORD	Operation count high word
		PumpData [{X}][12]	WORD	Operation count low word
		PumpData [{X}][13]	WORD	Idle time high word
		PumpData [{X}][14]	WORD	Idle time low word
AppliedPupSP	—	—	REAL	Setpoint given to each pump.
EngParam	—	—	FlowCTLGP_EngPar_DDT	Pump set engineering parameters.
	MinimumSP	—	REAL	Minimum pump setpoint.
	OptimalSP	—	REAL	Optimal pump setpoint.
	MaximumSP	—	REAL	Maximum pump setpoint.
	SwitchOnDelay	—	TIME	Delay between the end of the starting sequence of a pumping asset and the start of the starting sequence of another pumping asset.
	DrainingTime	—	TIME	Time for which Drain Valve is open when stopping the pumping asset.
	DelayAfterValveOpen	—	TIME	Delay time before next action after opening valves.
	DelayAfterValveClose	—	TIME	Delay time before next action after closing valves.
	DelayAfterPumpStart	—	TIME	Delay time before next action after starting pumps.
	DelayAfterPumpStop	—	TIME	Delay time before next action after stopping pumps.
	EnabledStrategies	—	INT	Determines strategies to be enabled. • 1 = Priority only • 2 = Balance only • 3 = Operation Count only • 4 = Idle Time only • 5 = Switch only • Any other value all = All enabled
	OutValveFirst	—	BOOL	1 = Outlet valve is opened before starting the pump and closed after stopping the pump.

NOTE:
- X ranges from 0 to 14.
- Y ranges from 1 to 15.

Control — FLOWASSET05GP

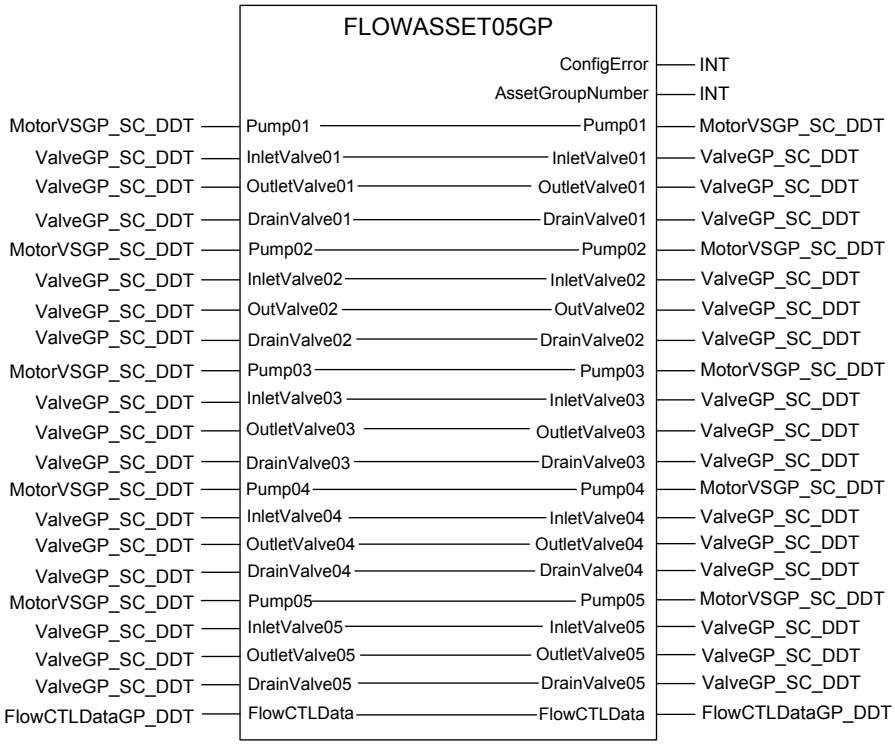
Overview

This chapter describes the FLOWASSET05GP DFB.

DFB Representation

Representation

The DFB that is used in the program has the following aspect at the section level. You can use it in any of the programming languages, although it is designed for use with the FBD language.



Outputs

Output Parameter Description

Parameter	Type	Description
ConfigError	INT	Configuration *error* detection code.
AssetGroupNumber	INT	Group number assigned as per the initialization sequence.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
Pump01	MotorGP_SC_DDT	Asset 01 pump
InletValve01	ValveGP_SC_DDT	Asset 01 inlet valve
OutletValve01	ValveGP_SC_DDT	Asset 01 outlet valve
DrainValve01	ValveGP_SC_DDT	Asset 01 drain valve
Pump02	MotorGP_SC_DDT	Asset 02 pump
InletValve02	ValveGP_SC_DDT	Asset 02 inlet valve

Parameter	Type	Description
OutletValve02	ValveGP_SC_DDT	Asset 02 outlet valve
DrainValve02	ValveGP_SC_DDT	Asset 02 drain valve
Pump03	MotorGP_SC_DDT	Asset 03 pump
InletValve03	ValveGP_SC_DDT	Asset 03 inlet valve
OutletValve03	ValveGP_SC_DDT	Asset 03 outlet valve
DrainValve03	ValveGP_SC_DDT	Asset 03 drain valve
Pump04	MotorGP_SC_DDT	Asset 04 pump
InletValve04	ValveGP_SC_DDT	Asset 04 inlet valve
OutletValve04	ValveGP_SC_DDT	Asset 04 outlet valve
DrainValve04	ValveGP_SC_DDT	Asset 04 drain valve
Pump05	MotorGP_SC_DDT	Asset 05 pump
InletValve05	ValveGP_SC_DDT	Asset 05 inlet valve
OutletValve05	ValveGP_SC_DDT	Asset 05 outlet valve
DrainValve05	ValveGP_SC_DDT	Asset 05 drain valve
PumpSetData	PumpSetData_GP_DDT	

NOTE: For additional information of those parameters that belong to `ValveGP_SC_DDT` and `MotorGP_SC_DDT` type.

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for flow control pattern.

Genies

Genie Properties

Refer to Genie Properties, page 62.

Genie Icons

Refer to Genie Icons, page 63.

Representation

Refer the genies of gpl_eqipmentmodule, page 862

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

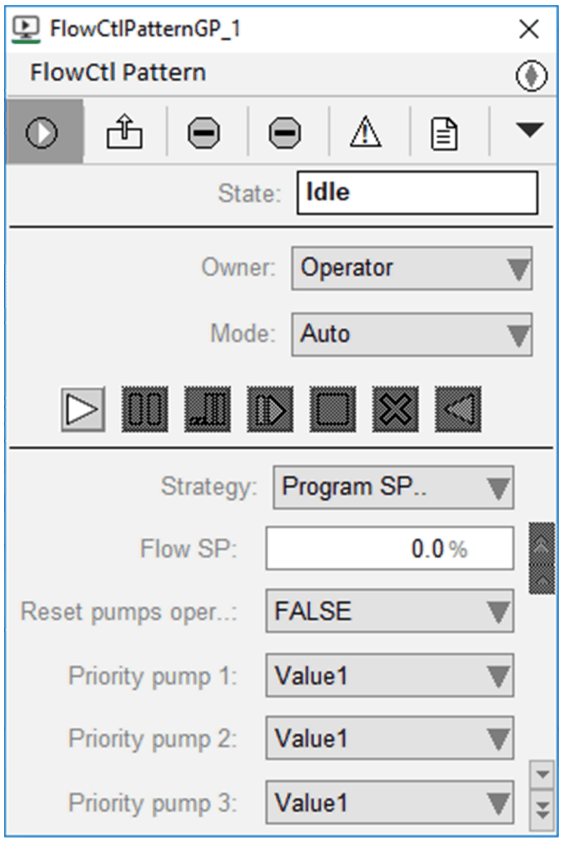
Available Tabs

During operation, clicking an flow control pattern genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Output Parameter
  - State Machine
  - Alarm, page 67
  - Event, page 69
- Tabs for optional functions, which appear only if configured:
  - InitCond, page 867
  - Failure, page 78

Operator Tab

The figure shows an operator tab:



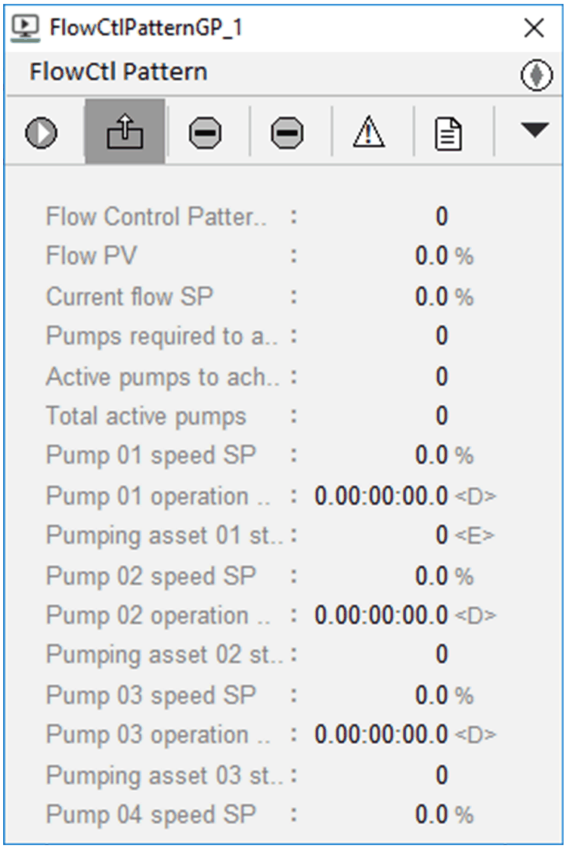
The table describes the control function of an operator tab:

Lable	Item Name	Description	Security parameter, page 818
State	State	State	–
Owner	OwnerSelect	Current owner of flow control pattern. For more details, refer to operating owner, page 58.	Set owner
Mode	Mode	Allows you to view and change the mode	Set auto mode, Set semi mode, set manual
Strategy	Strategy, OwnerSelect	Strategy of equipment module	Set strategy

Based on the strategy selection, reset pump operation, priority pumps and other combination will be available.

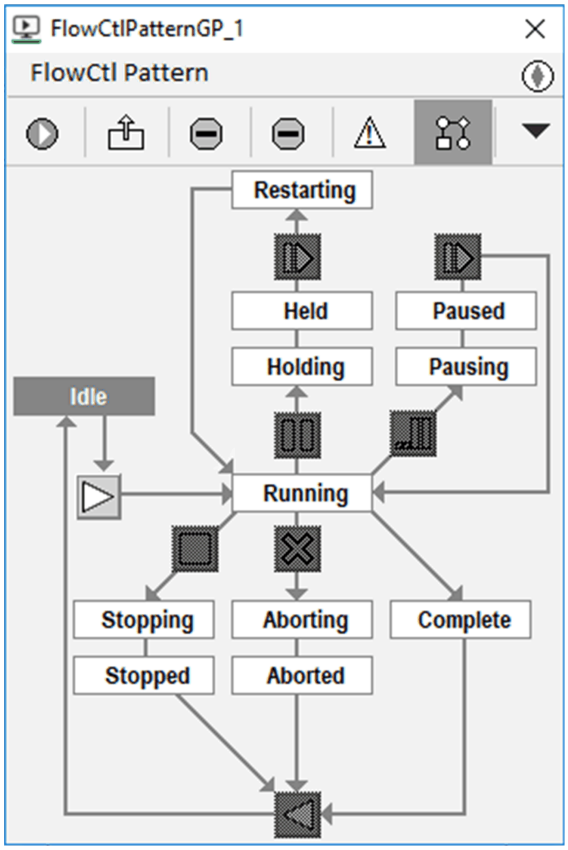
Output ParameterTab

The figure shows an output parameter tab:



State Machine Tab

The figure shows a state machine tab:



Items

Overview

This section describes the variables, equipstate variable, alarms, and events items of flow control pattern.

Variables

The table describes the variable items that are used by supervision components:

Item Name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`Transition{x}`	{Y} transition status	Done/ Pending	*{z}*
NOTE: {x} represents from 01 to 06. {y} represents from first to sixth. {z} represents from 0 to 5.			
`AutoResetDisable`	Auto reset disable status	Disabled/Not Disabled	*Bit15*
Bit items derived from ConfigurationWord.			
`OwnerSelect`	Owner of control module	Operator/ Program	*Bit0*
`SemiAutomatic`	Semi automatic mode command	SemiAuto-matic/Not	*Bit1*

Item Name	Description	Enumeration	Address
		Semi Automatic	
Manual	Equipment module sequence manual mode command	Auto/Manual	*Bit2*
NextStep	Equipment module sequence next step command	On/Off	*Bit3*
ScrollUp	Equipment module sequence scroll up command		*Bit4*
ScrollDown	Equipment module sequence scroll down command		*Bit5*
IO Items			
StatusWord	Status word	–	EMCTLG-P_ST.STW
ConfigurationWord	Configuration word		EMCTLG-P_ST.CFGW
Calculated items: Refer to calculated items of equipment module control, page 867			

Alarms

The table describes the advanced alarm items that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the state:

Item name	Description	Expressions
HeldForError	Held for error	GPL_DualBitCheck (State, 3,9,1)

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
StateAlarm	All changes in the equipment module pattern status and output.
ModeAlarm	All changes in the mode and owner.
ActionAlarm	All actions by the operator.

Auxiliary Functions of Flow Control Equipment Module

What's in This Chapter

PUMPINGASSETSEQ1GPGP 836

Overview

This section describes the function blocks that supplement the FLOWCTLPATTERN DFB and that have the objective of facilitating the programming of the equipment module.

PUMPINGASSETSEQ1GPGP

Overview

This section describes the PUMPINGASSETSEQ1GP DFB.

Description

General

It is an unprotected function block, that is used to control the start and stop sequence of a pumping asset. It evaluates the alert condition of the pumping asset diagnosis state.

Pumping Asset Sequence Execution

The Operation sequence to start a pumping asset is as follows:

- If used, close the Drain valve, wait for the Drain Valve to get closed (plus the delay time given by DelayAfterValveClose) or a detected failure.
- If OUTVALVEFIRST engineering parameter is True:
  - If used, open the Inlet valve, wait for the inlet valve to open (plus the delay time for each given by DelayAfterValveOpen) or for a detected failure.
  - If used, open the Outlet valve, wait for the outlet valve to open (plus the delay time for each given by DelayAfterValveOpen) or for a detected failure.
  - Start Pump, wait for the Pump running (plus the delay time given by DelayAfterPumpStart) or for a detected failure.
- If OUTVALVEFIRST engineering parameter is False:
  - If used, open the Inlet Valve, wait for the Inlet Valve to open (plus the delay time given by DelayAfterValveOpen) or for a detected failure.
  - Start Pump, wait for the Pump running (plus the delay time given by DelayAfterPumpStart) or for a detected failure.
  - If used, open the Outlet Valve, wait for the Outlet Valve to open (plus the delay time given by DelayAfterValveOpen) or for a detected failure.
- Wait for the delay time in case if it is needed to start more pumps.

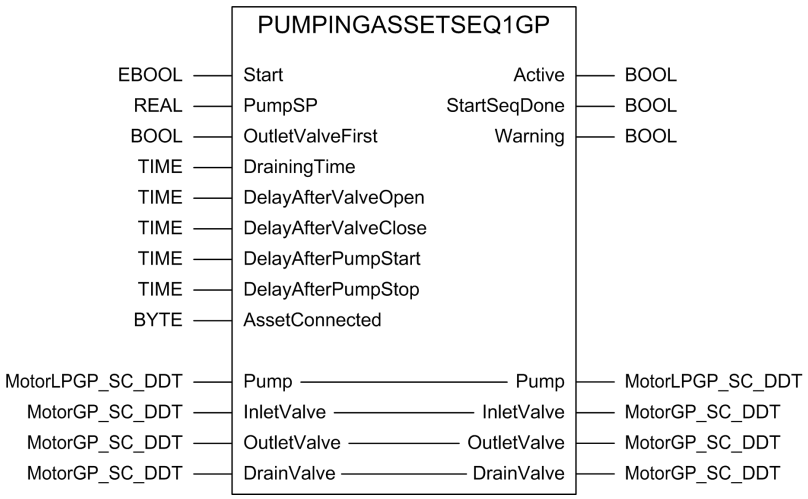
The Operation sequence to stop a pumping asset is as follows:

- If `OUTVALVEFIRST` engineering parameter is `True`:
  - Stop the pump, wait for the pump to stop (plus the delay time given by `DelayAfterPumpStop`) or for a detected failure.
  - If used, close the Inlet valve, wait for the inlet valve to close (plus the delay time for each given by `DelayAfterValveClose`) or for a detected failure.
  - If used, close the Outlet valve, and wait for the outlet valve to close (plus the delay time for each given by DelayAfterValveClose) or for a detected failure.
  - If used, open the Drain and wait for the Drain Valve to open (plus the delay time given by `DrainingTime`) or for a detected failure. Wait for the configured `DrainingTime` and then close the Drain Valve, wait for the Drain Valve to close (plus the delay time given by `DelayAfterValveClose`) or for a detected failure.
- Else, if `OUTVALVEFIRST` engineering parameter is `False`:
  - If used, close the Outlet Valve, wait for the Outlet Valve to get closed (plus the delay time given by `DelayAfterValveClose`) or a detected failure.
  - Stop the Pump, and wait for the Pump to stop (plus the delay time given by `DelayAfterPumpStop`) or for a detected failure.
  - If used, open the Drain valve, and wait for the Drain Valve to open (plus the delay time given by `DrainingTime`) or a detected failure. Wait for the configured `DrainingTime` and then close the Drain Valve, wait for the Drain Valve to close (plus the delay time given by `DelayAfterValveClose`) or for a detected failure.
  - If used, close the Inlet Valve, wait for the Inlet Valve to close (plus the delay time given by `DelayAfterValveClose`) or for a detected failure.

DFB Representation

Representation

The DFB that is used in the program has the following aspect at the section level. You can use it in any of the programming languages, although it is designed for use with the FBD language.



Inputs

Input Parameter Description

Parameter	Type	Description
Start	EBOOL	Trigger the start/stop operation.
PumpSp	REAL	Pump setpoint.
OutletValveFirst	BOOL	1 = Outlet valve is opened before starting the pump and closed after stopping the pump.
DrainingTime	TIME	Time for which drain valve will be open when stopping the pumping asset.
DelayAfterValveOpen	TIME	Delay time before next action after opening valves.
DelayAfterValveClose	TIME	Delay time before next action after closing valves.
DelayAfterPumpStart	TIME	Delay time before next action after starting pumps.
DelayAfterPumpStop	TIME	Delay time before next action after stopping pumps.
AssetConnected	BYTE	Indicates whether the components of a pumping asset are connected or not.

Outputs

Output Parameter Description

Parameter	Type	Description
Active	BOOL	Indicates the status of the pumping asset sequence: • True when start sequence is started. • False only when stop sequence is fully executed.
StartSeqDone	BOOL	Indicates that the start sequence has executed completely.
Warning	BOOL	Alerts based on each equipment availability/ detected failure/ alarm / owner / interlock.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
Pump	MotorVSGP_SC_DDT	Pump
InletValve	MotorGP_SC_DDT	Inlet valve
OutletValve	MotorGP_SC_DDT	Outlet valve
DrainValve	MotorGP_SC_DDT	Drain valve

NOTE: For additional information of those parameters that belong to MotorGP_SC_DDT type, page 148 and MotorVSGP_SC_DDT type, page 210.

Equipment Module

What's in This Chapter

Description of Equipment Module 839
Definition of Statuses, Equipment Module State and Commands 841
Strategy Execution Status Matrix 843
Strategy Execution Status Controller 844
Operating Modes 844
Modifying the EMPattern Template 845

Overview

This chapter describes the equipment module management.

Description of Equipment Module

Overview

The operation of the function block that is available for equipment module is described in this chapter.

The EMCTLGP DFB belongs to the General Purpose Library and is used to monitor and manage the number of specific minor processing activities (strategies). The module is based on the ISA-S88 Standard, Part 1 (S88.01) and implements the status controller defined in this standard for controlling procedure model components (phases, operations, and so on).

Function Description

User can use this DFB to change states in the process. In compliance with ISA-S88.01I, you can use this DFB in continuous processes to initiate start, stop or emergency sequences.

The DFB implements equipment state, strategy, strategy execution state management and commands received from:

- The Supervision system (when sequences are manually controlled by the operator)
- Other control sequences

The table compares the functions that are available in the DFB:

Function	EMCTL
Management of strategy execution state machine and commands according to ISA-S88	x
Management of messages for the operator integrated with the sequence	–
Management of operating modes according to ISA-S88: Manual (enables to change from one sequence step to any other sequence step), Semi (step advance with confirmation from the operator), and Auto	x
Owner Management: Operator/Program	x
Step time automatically calculated by the control block	x
Automatic management of transitions through the EMTRANSGP function block	x
Automatic management of sequence detected failure through the EMFAILEDGP function block	x
Automatic management of sequence steady-step marking through the EMSTABLESTEPGP function block	x

Function	EMCTL
Possibility of displaying the sequence graphically (active step and next level of transitions and steps) from the monitoring subsystem	x
NOTE: 'x' indicates that the functions are available in the DFB.	

Program the steps and transitions that the sequences implement (through ST or SFC language) while the component (EMCTL) allows the programmer to:

- Manage the statuses of the strategy execution sequence and
- Process the commands received from other levels as described above.

As a minimum, implement the strategy subsequence that implements normal operation (RUNNING.

Otherwise user can implement subsequences that attend to commands received from other levels of the control system or that react to detected failures in the process (HOLDING, RESTARTING, STOPPING, and ABORTING) detected by the strategy sequence itself.

The table describes the main functions of the DFB:

Function	Description
Processing strategy execution States and Commands	The DFB processes the commands received and determines the status of the strategy execution sequence with the objective of determining the subsequences (idle, hold, continue, abort, stop.) and which step needs to run in each cycle of the program.
Managing Entry Points in the strategy execution sequence	The DFB enables to retain the step number with which the execution of the normal sequence (Running) needs to be restarted (Restart) after being restarted from held status (Held).
Handling of Detected Failures	The component manages controller detected failures (without user intervention from the Supervision system). The component manages this detected failure and triggers the execution of the detected failure handling subsequence.
Strategy management	Strategy is any minor processing activity which an equipment module can perform, a maximum of 16 strategies are supported in equipment module. The DFB allows to select the strategy which need to be executed for the Equipment module. For an equipment module, strategy can be from 1 to 16. The Strategy codes and their interpretation is EM specific however the supervisory resources will allow the users to define associated text to the respected codes. The supervisory resource will not allow the user from modifying the Strategy while the status is not *IDLE*.
Equipment state management	The DFB allows to calculate the Equipment module state based on user defined logic. This DFB allows to configure up to 32 Equipment module states, where state 0 to 12 is as per recommendation defined in NS88 standard, 13 to 15 is reserved and 16 to 31 can be used to interpret user defined equipment state.
Disable strategy	The DFB allows to disable the strategy based on user defined logic so that disabled strategy will not be available in supervision for selection. Hence, the execution for disable strategy can be avoided.
Auto reset disable	Automatic reset (*STOPPED* → *IDLE*, automatically performs reset operation without consuming any extra execution cycle) can be disabled by setting the input pin (AUTORESETDIS) to a logical high.

Definition of Statuses, Equipment Module State and Commands

Statuses

The table describes the possible strategy execution statuses along with the description of each status:

Status	Description
Idle	Inactive operation. Waiting for command to start (**Start**) running.
Running	Normal operation.
Complete	Normal operation has finished. Requires an initialization order (**Reset**) to return to the Idle status.
Pausing	A pause (**Pause**) command is received for a short period while the normal operation was being executed (Running). The normal sequence runs until it automatically goes to paused (Paused) status.
Paused	Status reached after the pause sequence is completed and the next steady status is reached. Upon receiving the command to continue (**Resume**), the sequence continues with the next step of the normal operation sequence (Running).
Holding	A hold (**Hold**) command is received and the sequence that allows the process to be changed to a known status is executed. After this sequence is finished, it automatically goes to the held (Held) status.
HoldingFor-Error	A detected failure is automatically detected in the process (for example, a detected failure in Equipment Module), and the sequence that allows the process to be to a known status is executed. After this sequence is finished, it automatically goes to the held (Heldforerror) status.
Held	Status reached after the pause sequence is completed and the next steady status is reached.
HeldForError	Operation put on hold due to detected fail condition. Waiting for command to Restart (**Restart**) to get operation to running.
Restarting	A restart command (**Restart**) is received while the sequence is being held (Held). The sequence runs to return to the normal operation sequence (Running). The normal operation status (Running) is set automatically after the sequence is complete.
Stopping	An order to stop (**Stop**) has been received. The sequence to finish the sequence in a controlled way is executed (it is not possible to continue with it). The stopped operation status (Stopped) is automatically activated after the sequence is complete.
Stopped	The stop sequence (Stopping) is completed. Requires an initialization command (**Reset**) to go to inactive (Idle) status.
Aborting	An order to end (**Abort**) has been received. A fast stop, which is not necessarily controlled is executed. The aborted operation status (Aborted) is automatically activated after the sequence is completed.
Aborted	An aborting sequence (Aborting) has been completed. Requires an initialization command (**Reset**) to go to inactive (Idle) status.

The (Held) status can also be reached (without receiving a command) if the equipment module itself detects a problem situation. The DFB has a signal that allows this situation to be communicated.

The DFB has the necessary means to be able to restart the Running sequence from the last steady step in which the initial run was held by a **Hold** command.

Equipment Module State

The table describes the equipment module states along with codes and descriptions recommended by NS88 (ISA88 for Non-Stop production).

Equipment Module State	Code	Description
Off	0	The equipment is powered up, but is not yet ready to be operated.
Stopped	1	The equipment is ready to be started, is not performing its basic processing function, and equipment is not processing material.
Starting	2	The equipment has been commanded to start and is in the process of starting, but the equipment has not yet been commanded to process material.
Ready	3	The operation has finished starting and is ready for operation.
Standby	4	The equipment is running and is waiting for material to process.
Producing	5	The equipment is performing its basic processing function.
Switching	6	The equipment is finishing processing of a product and starting to process a new one.
Clearing	7	The equipment is finishing processing of the remaining product.
Holding	8	The equipment has been commanded to hold and stop processing and is performing the actions to go to held state.
Held	9	The equipment is not processing, it is waiting for resuming.
Stopping	10	The equipment has been commanded to stop processing and is performing to clear the material.
Aborting	11	The equipment has been commanded to abort (some emergency) processing and is performing to clear the material.
Aborted	12	The equipment has aborted processing and the material is cleared.
-	13...15	Reserved.
<User Defined>	16...31	These sixteen codes are reserved for general purpose states, so that the user can use as needed. In case of using them, the supervision resources have to be adjusted accordingly to interpret such codes and showing the right states to the operator in runtime.

Commands

The table describes the commands that can be communicated to the equipment module along with the description of each command:

Command	Description
Start	Allows the normal operation of the sequence (Running) to be started. It is only valid if the status is inactive (Idle).
Stop	Allows the sequence execution to be stopped (Running, Pausing, Paused, Holding, Held, or Restarting) and the stopping sequence to be run (Stopping).
Hold	Allows the sequence to be stopped (Running, Pausing, Paused, Holding, Held, or Restarting) and the stopping sequence to be run (Stopping).
Restart	Allows the sequence for continuing the operation (Restarting) and resume normal operation (Running).
Abort	Allows the sequence to be aborted (in any status except Idle, Complete, Aborting, and Aborted) and the aborting sequence to be run (Aborting).
Reset	Triggers the transition to idle status (in Complete, Aborted, or Stopped status).
Pause	Allows normal execution (Running) to be paused in the next steady Sequence status.
Resume	Allows normal operation of the sequence (Running) to be resumed from the paused (Paused) status.

Strategy Execution Status Matrix

Description

The following table describes the finite status controller that the DFB implements:

Initial State	Command	Activity	Final State (without command)
Idle	Start	Running	–
Running	Stop	Stopping	Stopped
	Hold	Holding	Held
	Pause	Pausing	Paused
	Abort	Aborting	Aborted
Complete	Reset	Idle	–
Pausing	Stop	Stopping	Stopped
	Hold	Holding	Held
	Abort	Aborting	Aborted
Paused	Stop	Stopping	Stopped
	Resume		Running
	Abort	Aborting	Aborted
Holding	Stop	Stopping	Stopped
	Abort	Aborting	Aborted
Held	Stop	Stopping	Stopped
	Restart	Restarting	Running
	Abort	Aborting	Aborted
Restarting	Stop	Stopping	Stopped
	Hold	Holding	Held
	Abort	Aborting	Aborted
Stopping	Abort	Aborting	Aborted
Stopped	Reset	Idle	–
	Abort	Aborting	Aborted
Aborting	No action possible		
Aborted	Reset	Idle	–

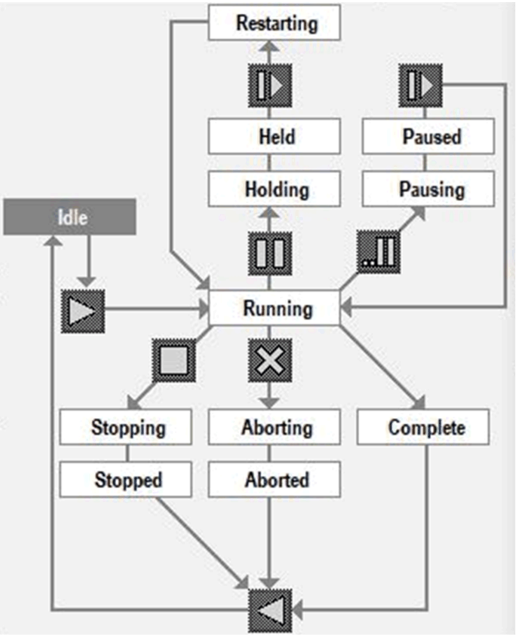
The `Held` status can also be reached (without receiving a command) if the equipment module itself encounters a detected error. The block has a signal that allows this situation to be communicated.

The block has the necessary means to be able to restart the Running sequence from the last steady step in which the initial run was held by a **Hold** command.

Strategy Execution Status Controller

Description

The following diagram describes the possible strategy execution statuses and commands that the Control Expert component manages according to the ISA-S88 standard:



Next to the transition arrows, the commands that the component receives (from the Supervision system or another higher level sequence) are shown in black, and status/commands that sequence obtains from other blocks (for example, user logic) are shown in parenthesis.

NOTE: This state transition diagram is derived from the first three initial states of the state transition .

Operating Modes

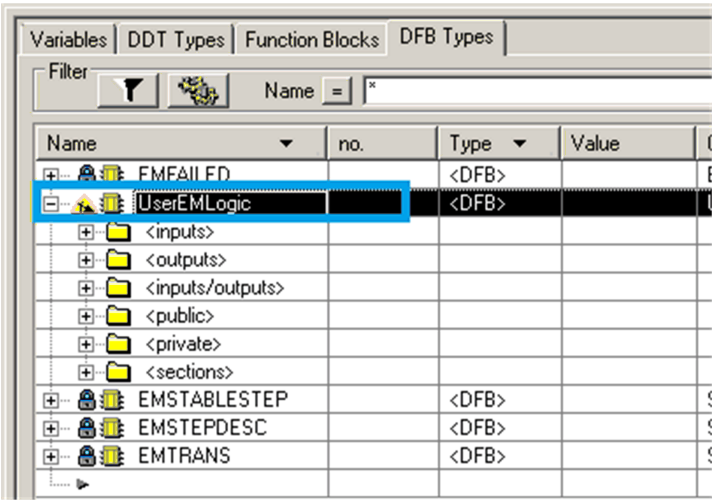
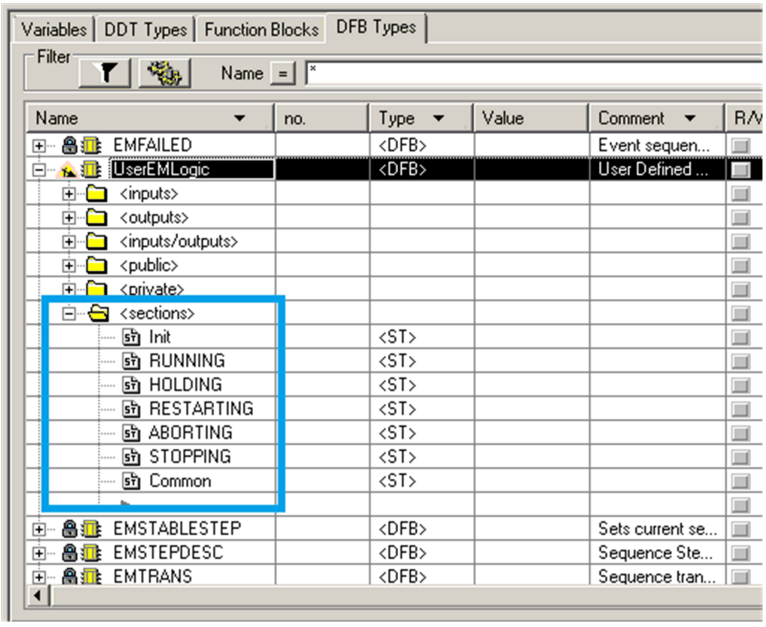
Mode Table

The following table provides the description of the sequence operating modes of `EMCTL` DFB:

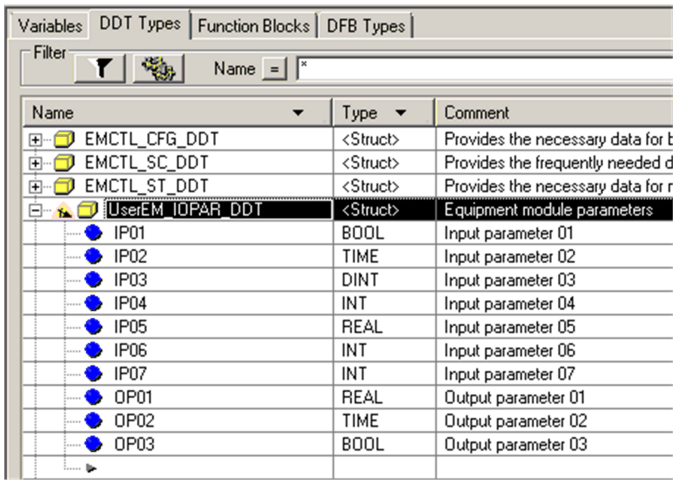
Operating mode	Implemented as:
Automatic	It is the normal running mode. The sequence of selected strategy runs automatically based on the transitions defined in the controller.
Semiautomatic	In this operating mode, after the conditions defined in the transitions are met, you need to confirm the move to the next step.
Manual	The sequence of selected strategy does not automatically advance to a step; user can manually change from one strategy execution step to another. Once the sequence step is selected, operating mode will change to semi-automatic and starts the strategy execution step selected in manual mode.

Modifying the EMPattern Template

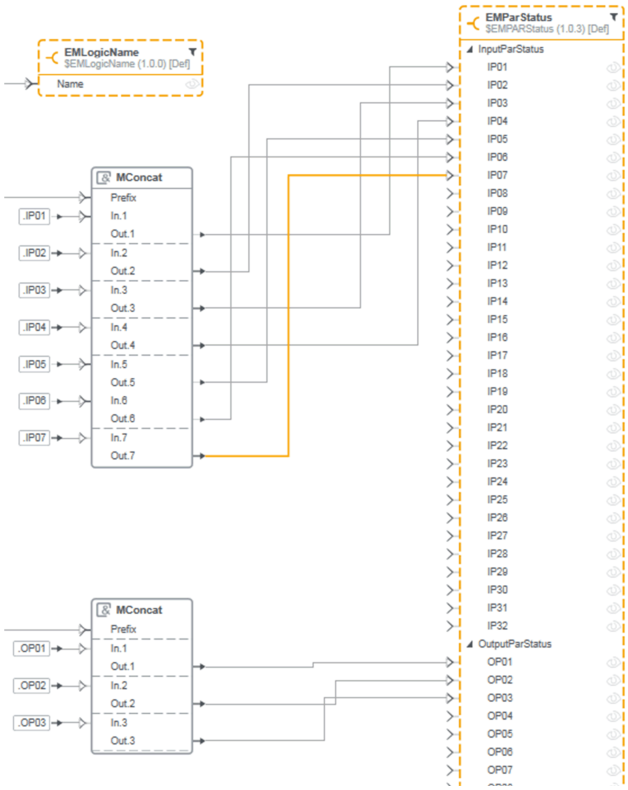
Steps to modify EMPattern template for the Equipment Module Operation

Step	Action
1	Navigate to the EMPattern template from **Global Templates**, this template is available under **General Purpose Library** folder with separate folder name as **Equipment Module Patterns**. **Global Templates** → **General Purpose Library** → **Equipment Module Patterns** → **Equipment Modules**.
2	You have to **Save As** the EMPattern template with different name (for example, UserEMPattern) so that the template can be used for specific needs.
3	You can write an own logic for specific requirement in the EMLogic_UL template in templatizer. Follow the path, UserEMPattern (newly created template from step 2) → EquipmentModulePattern_UC → EMLogic_UL).
4	Navigate to EMLogic_UL then go to templatizer, rename EMLogic (for example, UserEMLogic) DFB type as per the requirement.  You can import your EMLogic DFB into the templatizer. In this case if the DFB name is different than the existing name (EMLogic) then the links related to the DFB will be lost, you have to re-instantiate the new DFB and reconnect all the links to this DFB in the UL
5	You can write your own logic in the EMLogic DFB sections as shown below. 
6	It is recommended to rename the DDT Type EM_IOPAR_DDT (for example, UserEM_IOPAR_DDT), to avoid the DDT type conflict if you are using Equipment Module Pattern template to create two different templates with different number of input and output parameters. Also if you are importing EMLogic DFB in templatizer, then Verify that the DDT type name is different than EM_IOPAR_DDT.

Step	Action	
	Below example shows how parameters can be added and removed from the existing template. For the data type information of Input/Output parameters, page 856.	
	a	In templatizer, go to `UserEM_IOPAR_DDT`, add a new input parameter IP07 and remove the output parameter OP04 from DDT.



	b	After modifying the `EMLogic` DFB, input and output parameters in templatizer, analyze, build changes and save the templatizer.
6	c	Now in `EMLogic_UL`, add the new concatenation for input parameter IP07 and remove output parameter OP04 in `MConcat`. Link the newly added input parameter IP07 to `EMParStatus` as shown below.

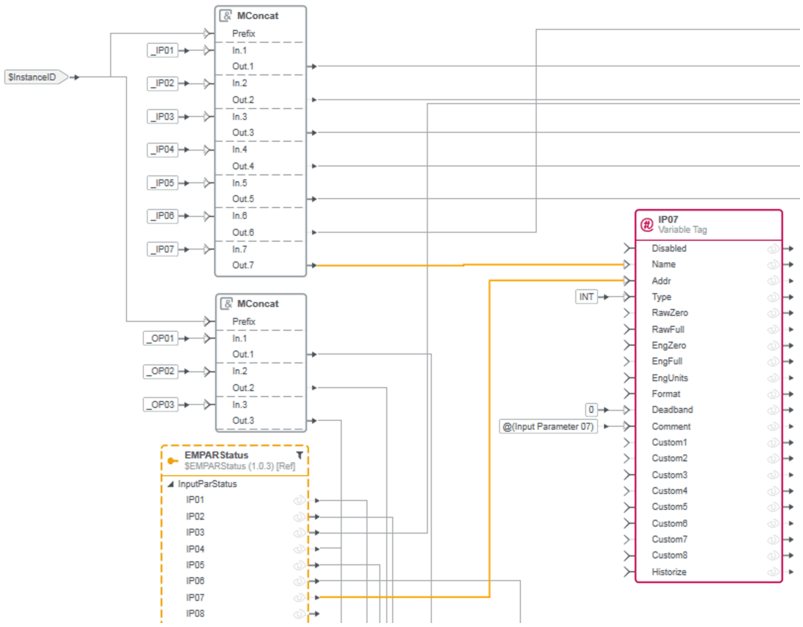
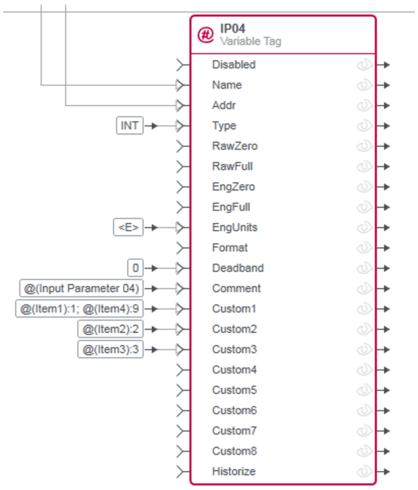


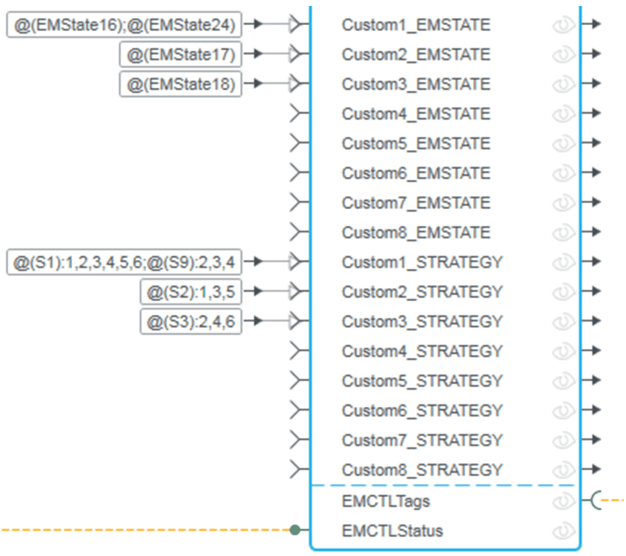
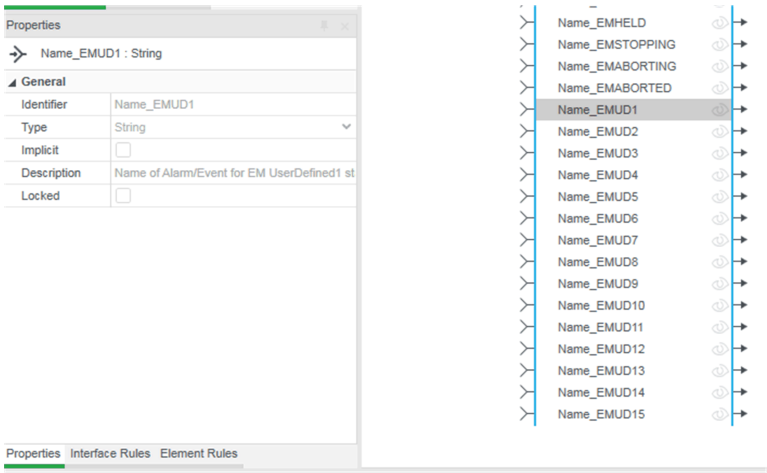
Verify the concatenated name **IP07** (with **.** as prefix) has to be same as defined in the `EM_IOPAR_DDT` type as mentioned in step 6 a.

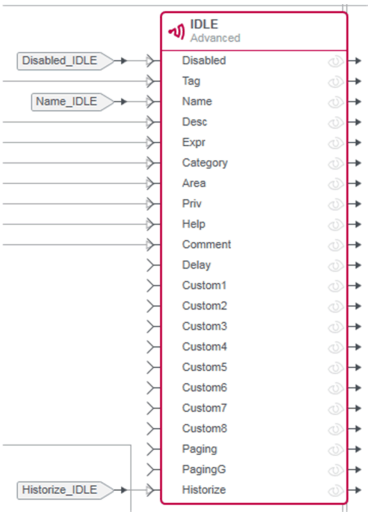
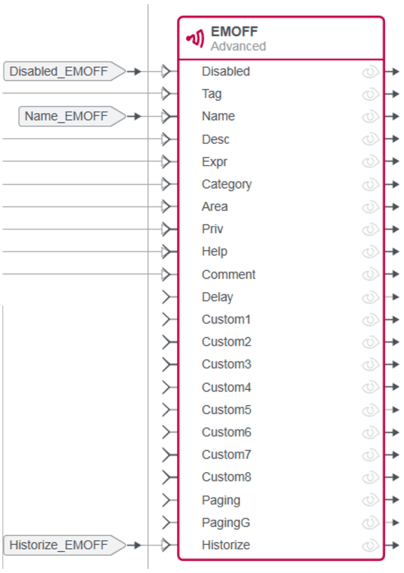
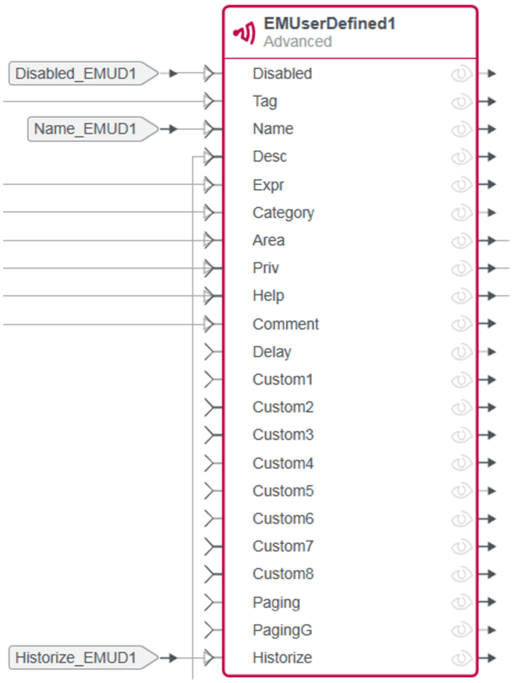
NOTE: To link a process template with Equipment module template, you have to export the data structure (DDT) of respective process template (for example, `MOTORGP_SC_DDT`) from the respective process template UL and import that DDT in templatizer and create a new pin and assign the respective DDT structure in data type.

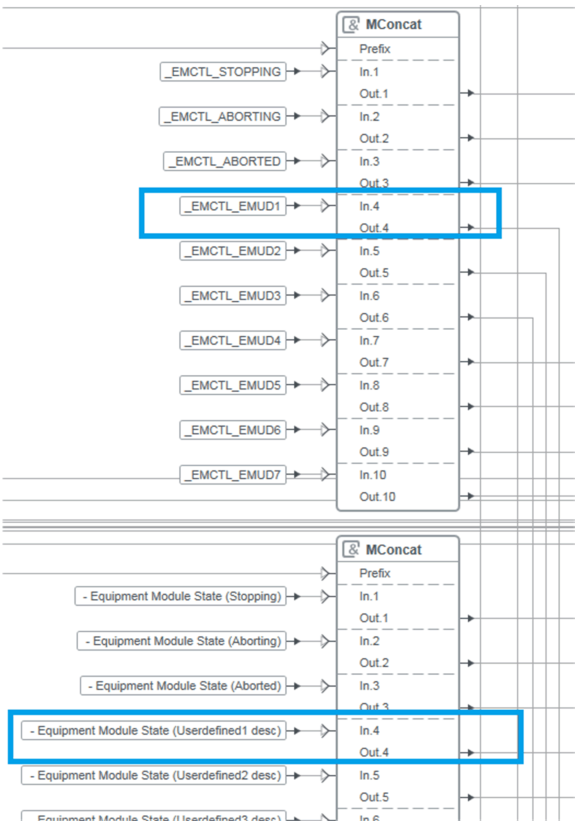
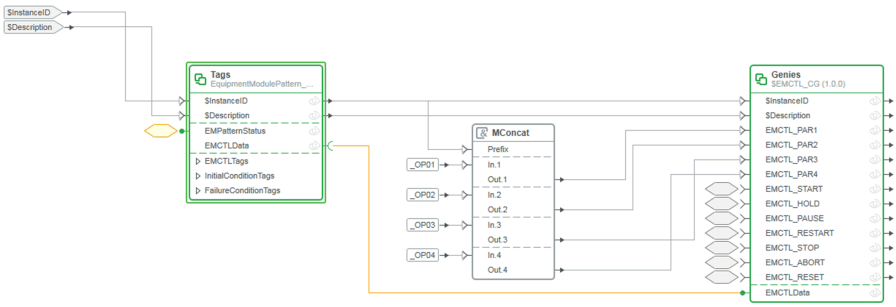
In UL, you have to instantiate the interface of corresponding process object (for example, `MOTORGP`) and link it to the respective DFB pin. Now you need to defer the interface to application level. Refer `PumpSetPattern` template for additional information.

6	d	After modifying the `EMLogic_UL`, **Save As** the template with a different name (for example, `UserEMLogic_UL`).

Step	Action	
	e	Open `EquipmentModulePattern_UC`, to replace the `EMLogic_UL` with newly created UL, drag and drop the newly created UL (`UserEMLogic_UL`) from **Browser** over the existing `EMLogic_UL`. Verify all the links with the existing `EMLogic_UL` are intact after replacing with new UL.
6	f	Now **Save As** the `EquipmentModulePattern_UC` with a different name (for example, `UserEquipmentModulePattern_UC`).
	g	Navigate and open `EMPAR_CD` template, `UserEMPattern` (newly created template from above point 2) → `EquipmentModulePattern_CS` → `EquipmentModulePattern_CD` → `EMPAR_CD`. Add the new variable tag with name IP07, add concatenation for IP07, link name and address, also provide the description and other parameters like range and units for the variable and remove the OP04 variable tag with all its references as shown in below figure. For the data type information of Input/Output parameters, page 856. 
6	h	To configure the input parameters as the Enumeration data type, you have to configure the **EngUnit** and **Custom** fields accordingly as shown. Format is, `<item name>:<item value>` (for example, `@(Item1):1;@(Item4):9`). EngUnit - `<E>` Below figure shows the example of Input parameter IP04 is configured for the Enumeration data type.  For additional information, refer to configuring the Enumeration data type, page 867.
	i	After modification, **Save As** the `EMPAR_CD` with a different name (for example, `UserEMPAR_CD`) and to replace the `EMPAR_CD` with newly created CD, drag and drop the newly created CD (`UserEMPAR_CD`) from **Browser** over the existing `EMPAR_CD` in `EquipmentModulePattern_CD`.
7	In `EquipmentModulePattern_CD` template, navigate to the `EMCTL_CD` and locate the `Custom` field parameters of `Strategy` and `EMState`, configure the `Strategy` and `EMState` by creating an input value for the element as shown.	

Step	Action
	Format is, `<Strategy text>:<list of applicable input parameters separated by a comma>` (for example, `@(S1):1,2,3,4,5,6;@(S9):2,3,4`). The Equipment Module states 0 to 15 are predefined and fixed. Equipment Module states 16 to 31 states are user defined states. To configure Equipment Module states you have to configure the `EMState` variable as `<State Description>;<State Description>` (for example, `@(EMState16);@(EMState24)`). For additional information, refer to configuring the `Strategy` and `EMState` management, page 867. 
8	You can Disable, Historize and configure the name of strategy execution state alarms in the **Application Explorer**. `Privilege`, `Help` and `Category` parameters are common for all the execution states and Equipment Module states. User can Disable, Historize and configure the name of Equipment Module state alarms from the `EMCTL_CD` by configuring its input values as per requirement. 
9	Parameters like `Delay`, `Custom1...Custom8`, `Paging` and `PagingG`) of Equipment Module states, Strategy execution states alarms has to be configured inside the `EMCTL_CD`. Below figure shows the parameters for Strategy execution state alarms.

Step	Action
	 Below figure shows the parameters for predefined EMstates alarms. 
9	Below figure shows the parameters for user defined Equipment Module state alarms. 

Step	Action
10	While configuring the user defined Equipment Module states you can rename the parameters and `EMstate` alarm variable names as per requirement. You can modify the tag name and description of the user defined Equipment Module states.  **NOTE:** You have to Verify that the predefined Equipment Module state alarm variables, tag names and descriptions are not to be modified. You can **Defer** required configuration parameters to Equipment ModuleEquipment Module level so that it can be available for modification in **Application Explorer**. If certain alarm tags are not used, then those tags can be deleted in `EMCTL_CD`.
11	After modification, **Save As** the `EMCTL_CD` template with a different name (for example, `UserEMCTL_CD`) and to replace the `EMCTL_CD` with newly created CD, drag and drop the newly created CD (`UserEMCTL_CD`) from **Browser** over the existing `EMCTL_CD` in `EquipmentModulePattern_CD`.
12	Now **Save As** the `EquipmentModulePattern_CD` with a different name (for example, `UserEquipmentModulePattern_CD`) and to replace the `EquipmentModulePattern_CD` with newly created CD, drag and drop the newly created CD (`UserEquipmentModulePattern_CD`) from **Browser** over the existing `EquipmentModulePattern_CD` in `EquipmentModulePattern_CS`.
13	To configure genie parameters DATA1 to DATA4, you have to go to `EquipmentModulePattern_CS` and concatenate the input/output parameter tag as per requirement as shown below. The figure shows the output parameter1 to output parmeter4 that are configured for DATA1 to DATA4. 
14	Now save as the `EquipmentModulePattern_CS` with different name (for example, `UserEquipmentModulePattern_CS`). Now open and to replace the `EquipmentModulePattern_CS` with newly created CS, drag and drop the newly created CS (`UserEquipmentModulePattern_CS`) from **Browser** over the existing `EquipmentModulePattern_CS` in `EMPattern` .
15	Open `UserEMPattern` template (created from step 2) to replace UC and CS with the newly created UC and CS. Now, to replace the `EquipmentModulePattern_CS` with newly created CS, drag and drop the newly created CS (`UserEquipmentModulePattern_CS`) from **Browser** over the existing `EquipmentModulePattern_CS`.

Step	Action
	To replace the `EquipmentModulePattern_UC` with newly created UC (`UserEquipmentModulePattern_UC` created from step 6 f), drag and drop the newly created UC (`UserEquipmentModulePattern_UC`) from **Browser** over the existing `EquipmentModulePattern_UC`
16	Close all the open templates (if any). Go to **Global Templates** and navigate to the newly created main template `UserEMPattern`. Right click and **Update**.
17	Verify that this newly created template is updated with the all user modified composite templates. Now, the template is ready to use.

EMCTLGP - Equipment Module Controller

What's in This Chapter

Template 852
Control 855
Supervision 862

Overview

This chapter describes the components that provide the functionality of template, control services, and Supervision functions of the EMCTLGP DFB.

Template

Overview

This section describes the functionality, parameters, and interfaces of the $EMPatternGP template.

Description

General Description

The $EMPatternGP template is used to monitor and manage equipment module patterns.

The composite and facet names without the **$** prefix can be modified, while the composite and facet names with **$** prefix are standard and cannot be modified.

These templates do not reflect any specific installation.

Parameters

Parameters

The $EMPatternGP template defines operation of a typical equipment module pattern. Alternative equipment module pattern facets enable the modification of the equipment module pattern that are available.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Disable auto reset	Boolean	False	–
Description	**InitialConditions**			
	Condition 1 to 15	String	–	–
	Group description		@(Initial Conditions)	–
	FailureConditions			

Section	Parameter description	Type	Default	Additional remarks
	Condition 1 to 15	String	–	–
	Group description		@ (Fail-ures)	–
Reset Required	**InitialConditions**			
	Condition 1 to 15	Boolean	False	–
	FailureConditions			
	Condition 1 to 15	Boolean	False	–
Disable Bypass	**InitialConditions**			
	Condition 1 to 15	Boolean	False	–
	FailureConditions			
	Condition 1 to 15	Boolean	False	–

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Operation	Normal owner	Enum	Program (1)	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	Set owner	Enum	Operator (1)	Privilege to change owner. This parameter is used in Operator tab, page 864.
	Set alarm acknowledge		Operator (1)	Privilege to set alarm acknowledge. This parameter is used in Alarm tab, page 67.
	Set semi mode		Operator (1)	Privilege to set semi mode. This parameter is used in Operator tab, page 864.
	Set manual mode			Privilege to set manual mode. This parameter is used in Operator tab, page 864.
	Set next step			Privilege to set next step. This parameter is used in Operator tab, page 864.
	Set scroll up			Privilege to set scroll up. This parameter is used in Operator tab, page 864.
	Set scroll down			Privilege to set scroll down. This parameter is used in Operator tab, page 864.
	Set start			Privilege to set start. This parameter is used in Operator tab, page 864.
	Set hold			Privilege to set hold. This parameter is used in Operator tab, page 864.
	Set restart			Privilege to set restart. This parameter is used in Operator tab, page 864.
	Set stop			Privilege to set stop. This parameter is used in Operator tab, page 864.
	Set reset		Operator (Confirmed) (10)	Privilege to reset. This parameter is used in Operator tab, page 864.
	Set pause	Enum	Operator (1)	Privilege to set pause. This parameter is used in Operator tab, page 864.
	Set resume			Privilege to set resume. This parameter is used in Operator tab, page 864.
	Set abort			Privilege to set abort. This parameter is used in Operator tab, page 864.
	Set strategy			Privilege to set strategy. This parameter is used in Operator tab, page 864.

Asset and Display Parameters

For more details, refer to the topic describing the parameters of asset and display, page 43.

Data

The table describes the data parameters:

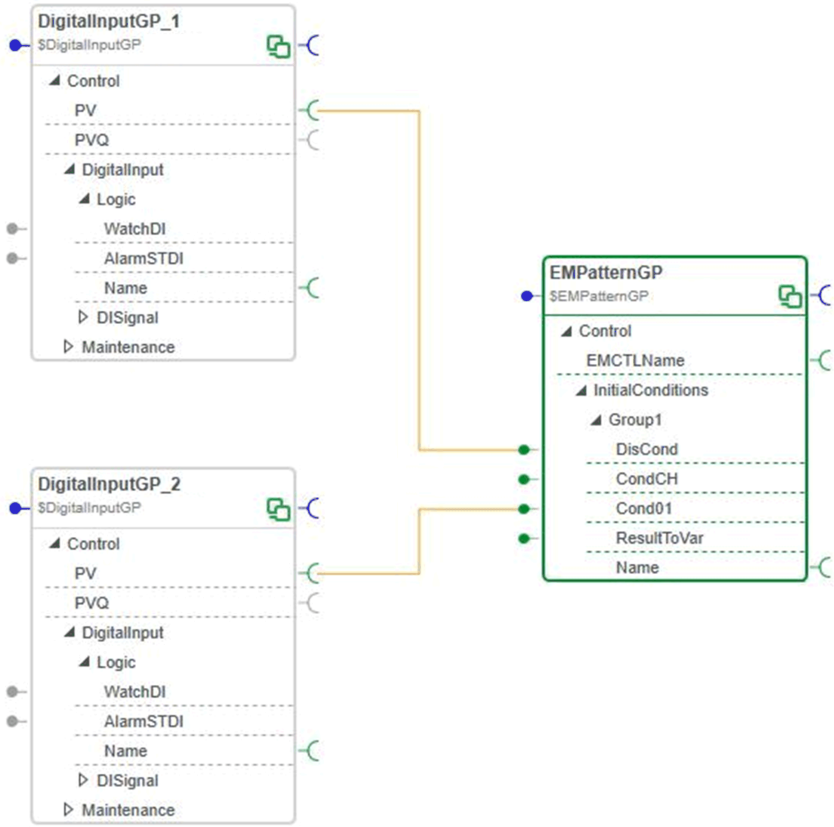
Section	Parameter description	Type	Default	Additional remarks
Configuration	Disable start command button	Boolean	False	–
	Disable hold command button			–
	Disable pause command button			–
	Disable reset command button			–
	Disable stop command button			–
	Disable abort command button			–
	Disable restart command button			–
Alarm	Held for error description	String	@(Held For Error)	–
	Held for error severity	Enum	Low (4000)	–
	Held for error group	String	Failure	–
Event	Enable log	Boolean	False	–
Historize	Execution state	Boolean	False	–
	Current step description			–
	Step description			–
	Next step			–
	Transition description			–
	Disable strategy			–
	Equipment module state			–
	Equipment state			–
	Sequence equipment module mode			–
	Strategy			–
	Sequence command			–
	Elapsed time			–
Historize	**InitalConditionTags** and **Failure Conditions Tags**			
	Condition	Boolean	False	–

Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

This figure shows the $EMPatternGP (also applicable for $EMPatternCE) template as it appears in Links Editor and an example of other templates connected to it:



For interface identifiers, refer to the control module template interfaces, page 49.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

Overview

This chapter describes the EMCTLGP DFB.

Description

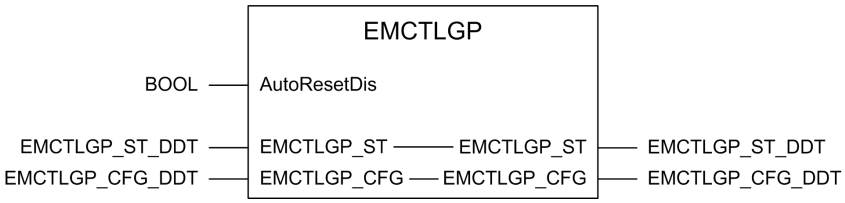
Initialization

If the strategy execution status is unknown, it is initialized as Idle regardless of whether a controller has restarted or not. This initialization is done automatically.

DFB Representation

Representation

The DFB that is used in the program has the following aspect at the section level. You can use it in any of the programming languages, although it is designed for use with the FBD language.



Inputs

Input Parameter Description

Parameter	Type	Description
AutoResetDis	BOOL	Disabling the automatic reset of the strategy execution state.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
EMCTLGP_ST	EMCTLGP_ST_DDT	Provides the necessary data for monitoring and controlling equipment module.
EMCTLGP_CFG	EMCTLGP_CFG_DDT	Provides the necessary data for background monitoring.

EMCTLGP_ST_DDT Type

Name	Bit Name	Bit	Type	Description
ETime	-	-	UDINT	The DFB calculates the execution time elapsed in tenths of a second for the current step of the control sequence. If execution of the program is stopped and resumed, the accumulated execution time is considered but not the time that the program was stopped.
Strategy	-	-	UINT	Selected strategy in equipment module.
DISStrategy	-		WORD	Disabling the execution of strategy.
	DISStrategy01	Bit0	BOOL	Disabling the execution of strategy 1.
	DISStrategy02	Bit1	BOOL	Disabling the execution of strategy 2.
	...	...	...	...
	DISStrategy16	Bit15	BOOL	Disabling the execution of strategy 16.
EMState	-	-	UINT	Equipment module state. For additional information refer Equipment state management, page 841.
State	-	-	UINT	Read-only access

Name	Bit Name	Bit	Type	Description
				Provides the status of the strategy execution. Read-only access to the data is contained in this bits word. Refer to the Definition of Statuses, Equipment Module State and Commands, page 841.
	Idle	Bit0	BOOL	Inactive operation. Waiting for command to start (Start) running.
	Running	Bit1	BOOL	Normal operation.
	Held	Bit2	BOOL	Status reached after the pause sequence is completed and the next steady state is reached.
	HeldForError	Bit3	BOOL	Status reached after the pause sequence is completed and the next steady state is reached to a detected failure.
	Stopped	Bit4	BOOL	The stop sequence (Stopping) is completed. Requires an initialization command (Reset) to go to inactive (Idle) status.
	Paused	Bit5	BOOL	Status reached after the pause sequence is completed and the next steady status is reached. Upon receiving the command to continue (Resume), the sequence continues with the next step of the normal operation sequence (Running).
	Aborted	Bit6	BOOL	An aborting sequence (Aborting) has been completed. Requires an initialization command (Reset) to go to inactive (Idle) state.
	Complete	Bit7	BOOL	Normal operation has finished. Requires an initialization order (Reset) to return to the Idle status.
	Holding	Bit8	BOOL	A hold (Hold) command is received and the sequence that allows the process to be changed to a known status is executed. After this sequence is finished, it automatically goes to the held (Held) status.
	HoldingForEr-ror	Bit9	BOOL	A detected failure is automatically detected in the process (for example, a detected failure in Equipment Module), and the sequence that allows the process to be to a known status is executed. After this sequence is finished, it automatically goes to the held (HeldForError) status.
State	Restarting	Bit10	BOOL	Restart command (Restart) is received while the sequence is being held (Held). The sequence runs to return to the normal operation sequence (Running). The normal operation status (Running) is set automatically after the sequence is complete.
	Pausing	Bit11	BOOL	A pause (Pause) command is received for a short period while the normal operation was being executed (Running). The normal sequence runs until it automatically goes to paused (Paused) status.
	Stopping	Bit12	BOOL	An order to stop (Stop) has been received. The sequence to finish the sequence in a controlled way is executed (it is not possible to continue with it). The stopped operation status (Stopped) is automatically activated after the sequence is complete.
	Aborting	Bit13	BOOL	An order to end (Abort) has been received. A fast stop, which is not necessarily controlled is executed. The aborted operation status (Aborted) is automatically activated after the sequence is completed.
Command	-	-	UINT	Read-only access Enables commands to be sent to the strategy execution sequence usually from the Supervision system. Write access to the data is contained in this bits word. The DFB sets the command to 0 after processed (successfully or not). Refer to the Definition of States, Equipment Module State and Commands, page 841.
	Start	Bit0	BOOL	Allows the normal operation of the sequence (Running) to be started.
	Hold	Bit1	BOOL	Allows the sequence to be stopped (Running, Pausing, Paused, Holding, Held, or Restarting) and the stopping sequence to be run (Stopping).
	Restart	Bit2	BOOL	Allows the sequence for continuing the operation (Restarting) and resume normal operation (Running).
	Stop	Bit3	BOOL	Allows the sequence to be stopped (Running, Pausing, Paused, Holding, Held, or Restarting) and the stopping sequence to be run (Stopping).
	Reset	Bit4	BOOL	Triggers the transition to idle status (in Complete, Aborted, or Stopped status).

Name	Bit Name	Bit	Type	Description
	Pause	Bit5	BOOL	Allows normal execution (Running) to be paused in the next steady Sequence status.
	Resume	Bit6	BOOL	Allows normal operation of the sequence (Running) to be resumed from the paused (Paused) status.
	Abort	Bit7	BOOL	Allows the sequence to be aborted (in any status except Idle, Complete, Aborting, and Aborted) and the aborting sequence to be run (Aborting).
STW	-	-	WORD	Read-only access Sequence status word. In this word, the *EMTRANS* auxiliary DFB is used to load the status of the transitions (maximum of six) from the current step of the sequence to the steps in the next level of the sequence, where 0 means that the conditions in the transition are not fulfilled and 1 means they are fulfilled.
	Status1Transition	Bit0	BOOL	Status of the first transition
	Status2Transition	Bit1	BOOL	Status of the second transition
	Status3Transition	Bit2	BOOL	Status of the third transition
	Status4Transition	Bit3	BOOL	Status of the fourth transition
	Status5Transition	Bit4	BOOL	Status of the fifth transition
	Status6Transition	Bit5	BOOL	Status of the sixth transition
	AutoResetDis	Bit15	BOOL	Auto reset disable
CFGW	-		WORD	Sequence configuration word.
	Owner	Bit0	BOOL	Read/write access Enables to configure whether the sequence commands will come from the program (0) or the operator (1).
	Semi	Bit1	BOOL	Read/write access Enables to configure the sequence in automatic (0) or semi-automatic (1) mode.
	Manual	Bit2	BOOL	Read/write access Enables to configure the strategy execution in automatic/semi-automatic (0) or manual (1) mode. **Semi** / **Manual** / Mode OFF / OFF / Automatic OFF / ON / Semi-automatic ON / – / Manual
	NextStep	Bit3	BOOL	Read/write access Enables to confirm the change of steps to the next sequence step in Semi-Automatic mode. The DFB sets the signal to 0 after it is processed.
	ScrollUp	Bit5	BOOL	Read/write access Enables to show the steps previous to the one being shown on the current window in manual mode. The DFB sets the signal to 0 after it is processed.
	ScrollDown	Bit6	BOOL	Read/write access Enables to show the steps following the one being shown on the current window in manual mode. The DFB sets the signal to 0 after it is processed.
CStepD	-	-	String [122]	Current step description (three characters for step number concatenated with seventeen characters for description).

EMCTLGP_CFG_DDT Type

Name	Type	Description
NStep	INT	Number of the new step at which the subsequence needs to be positioned in manual mode. The DFB sets this variable to 0 after it is processed.
StepD	STRING[122]	Description of steps. Limited to six steps. In automatic and semi-automatic mode, it shows the destination steps specified in the current step transitions. In manual mode, it shows the list of the steps defined in the strategy execution sequence. This variable format is: six steps, twenty characters each. Of the twenty characters of each step, the first three are for the step number and the next seventeen are for its description. The value of this variable is assigned for the auxiliary functions.
TransD	STRING[122]	Description of transitions. Limited to six transitions. This variable format is six transitions, twenty characters each. The twenty characters are for its description. The value of this variable is assigned for the auxiliary functions.

Public Variables

Public Variable Description

Variable	Type	Description
SC	EMCTLGP_SC_DDT	Provides the frequently needed data to monitor the equipment module status and control it from the sequential control.

EMCTLGP_SC_DDT Type

Name	Bit Name	Bit	Type	Description
ETime	-	-	UDINT	Elapsed time in the current step (dsec).
Strategy	-	-	UINT	Read-only access Current strategy of equipment module.
EMState	-	-	UINT	Read-only access Equipment module state.For addition information refer Equipment state management, page 839.
State	-	-	UINT	Read-only access Refer to the *EMCTLGP_ST.State* input/output pin, page 856.
	Idle	Bit0	BOOL	Inactive operation. Waiting for command to start (Start) running.
	Running	Bit1	BOOL	Normal operation.
	Held	Bit2	BOOL	Status reached after the pause sequence is completed and the next steady state is reached.
	HeldForError	Bit3	BOOL	Status reached after the pause sequence is completed and the next steady state is reached to a detected failure.
	Stopped	Bit4	BOOL	The stop sequence (Stopping) is completed. Requires an initialization command (Reset) to go to inactive (Idle) status.
	Paused	Bit5	BOOL	Status reached after the pause sequence is completed and the next steady status is reached. Upon receiving the command to continue (Resume), the sequence continues with the next step of the normal operation sequence (Running).
	Aborted	Bit6	BOOL	An aborting sequence (Aborting) has been completed. Requires an initialization command (Reset) to go to inactive (Idle) state.
State	Complete	Bit7	BOOL	Normal operation has finished. Requires an initialization order (Reset) to return to the Idle status.

Name	Bit Name	Bit	Type	Description
	Holding	Bit8	BOOL	A hold (Hold) command is received and the sequence that allows the process to be changed to a known status is executed. After this sequence is finished, it automatically goes to the held (Held) status.
	HoldingForError	Bit9	BOOL	A detected failure is automatically detected in the process (for example, a detected failure in Equipment Module), and the sequence that allows the process to be to a known status is executed. After this sequence is finished, it automatically goes to the held (HeldForError) status.
	Restarting	Bit10	BOOL	Restart command (Restart) is received while the sequence is being held (Held). The sequence runs to return to the normal operation sequence (Running). The normal operation status (Running) is set automatically after the sequence is complete.
	Pausing	Bit11	BOOL	A pause (Pause) command is received for a short period while the normal operation was being executed (Running). The normal sequence runs until it automatically goes to paused (Paused) status.
	Stopping	Bit12	BOOL	An order to stop (Stop) has been received. The sequence to finish the sequence in a controlled way is executed (it is not possible to continue with it). The stopped operation status (Stopped) is automatically activated after the sequence is complete.
	Aborting	Bit13	BOOL	An order to end (Abort) has been received. A fast stop, which is not necessarily controlled is executed. The aborted operation status (Aborted) is automatically activated after the sequence is completed.
Command	-	-	UINT	Read/write access Sequence commands.
	Start	Bit0	BOOL	Allows the normal operation of the sequence (Running) to be started.
	Hold	Bit1	BOOL	Allows the sequence to be stopped (Running, Pausing, Paused, Holding, Held, or Restarting) and the stopping sequence to be run (Stopping).
	Restart	Bit2	BOOL	Allows the sequence for continuing the operation (Restarting) and resume normal operation (Running).
	Stop	Bit3	BOOL	Allows the sequence to be stopped (Running, Pausing, Paused, Holding, Held, or Restarting) and the stopping sequence to be run (Stopping).
	Reset	Bit4	BOOL	Triggers the transition to idle status (in Complete, Aborted, or Stopped status).
	Pause	Bit5	BOOL	Allows normal execution (Running) to be paused in the next steady Sequence status.
	Resume	Bit6	BOOL	Allows normal operation of the sequence (Running) to be resumed from the paused (Paused) status.
	Abort	Bit7	BOOL	Allows the sequence to be aborted (in any status except Idle, Complete, Aborting, and Aborted) and the aborting sequence to be run (Aborting).
	Completed	Bit8	BOOL	1 = Sends a command to finish the current subsequence (Running, Holding, Restarting, Stopping, and so on). 0 = The signal is automatically set to 0 by the DFB after the signal is processed. As an alternative, forcing the value of this variable to 1 by using the EMTRANSGP auxiliary DFB is recommended.
	StableStep	Bit9	BOOL	1 = Sends a command to mark the current step as a steady step, that is: • The step with which the sequence will resume after its is held or • The step on which the sequence will be held after the sequence is paused. 0 = The signal is automatically set to 0 by the DFB after the signal is processed. As an alternative, forcing the value of this variable to 1 by using the EMSTABLESTEPGP auxiliary DFB is recommended.

Name	Bit Name	Bit	Type	Description
CStep	-	-	INT	Number of the active sequence current step (Running, Pausing, Holding, Restarting, Stopping, or Aborting). The value 0 corresponds to the initial step of each subsequence.
NextStep	-	-	INT	Enables the number of the next step to be run in the sequence to be set. The signal is automatically set to 0 by the DFB after the signal is processed. As an alternative, forcing the value of this variable to 1 by using the EMTRANSGP auxiliary DFB is recommended.
Mode	-	-	INT	Mode is encoded as (bits 1 and 2): • 0 = Automatic • 1 = Semi-Automatic • 2 = Manual
Currentitem	-	-	INT	1 = Indicates the current position of the list of steps in Manual mode. The value –1 indicates that there has been a mode change and -2 indicates that the mode change has been identified.
Dest (Dest[1] ... Dest[6]	-	-	ARRAY [1...6] of INT	Destinations steps after transitions.
DestW	-	-	WORD	Transitions Conditions word.
Properties	-	-	WORD	Collective word of necessary boolean variables.
	Owner	Bit0	BOOL	Read-only access Enables to configure whether the sequence commands will come from the program (0) or the operator (1).
	OnEntry	Bit1	BOOL	Read-only access 1 = Reports that a new sequence step has just been started. User can use this event in the logic of the sequence step actions to run any actions. The actions run only when the step starts.
	OnExit	Bit2	BOOL	Read-only access 1 = Reports that the execution of the sequence current step is about to finish. User can use this event in the logic of the sequence step actions to run any actions. The actions run only when the step ends.
	Fail	Bit4	BOOL	Read/write access 1 = Sends a command to stop the sequence normal execution. 0 = The signal is automatically set to 0 by the DFB after the signal is processed. With the activation of this signal, the Holding For Detected Error subsequence is run next if the Running, Pausing, Restarting, or Paused subsequence was running previously. As an alternative, forcing the value of this variable to 1 by using the EMFAILEDGP auxiliary DFB is recommended.
	NewCycle	Bit5	BOOL	Read-only access 1 = Indicates that a new sequence state machine execution cycle has started.
	UpdateStepD	Bit6	BOOL	Read-only access 1= Internal mark indicates the need for updating the steps descriptions.
Properties	RunningX	Bit7	BOOL	Read-only access 1 = Indicates that the current state of the sequence is Running.
	HoldingX	Bit8	BOOL	Read-only access 1 = Indicates that the current state of the sequence is Holding or Holding for Detected Error.

The following elements (status) are described in the Status Controller section, page 844. The Definition of Statuses, Equipment Module State and Commands, page 841 shows the sequence finite state controller.

Status	Type	Description
Idle	BOOL	1 = Indicates that the current state of the sequence is Idle.
Running	BOOL	1 = Indicates that the current state of the sequence is Running.
Complete	BOOL	1 = Indicates that the current state of the sequence is Complete.
Pausing	BOOL	1 = Indicates that the current state of the sequence is Pausing.
Paused	BOOL	1 = Indicates that the current state of the sequence is Paused.
Holding	BOOL	1 = Indicates that the current state of the sequence is Holding.
HoldingFor-Error	BOOL	1 = Indicates that the current state of the sequence is `Holding` or `Holding for Detected Error`.
Held	BOOL	1 = Indicates that the current state of the sequence is Held.
HeldForEr-ror	BOOL	1 = Indicates that the current state of the sequence is Held for detected error.
Restarting	BOOL	1 = Indicates that the current state of the sequence is Restarting.
Stopping	BOOL	1 = Indicates that the current state of the sequence is Stopping.
Stopped	BOOL	1 = Indicates that the current state of the sequence is Stopped.
Aborting	BOOL	1 = Indicates that the current state of the sequence is Aborting.
Aborted	BOOL	1 = Indicates that the current state of the sequence is Aborted.

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the equipment module control.

Genies

Genie Properties

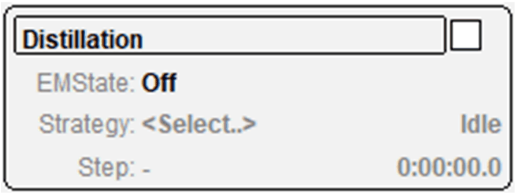
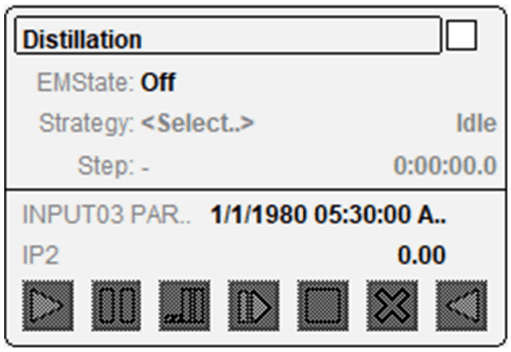
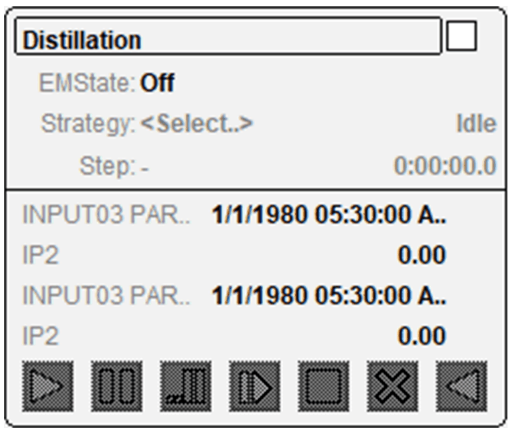
Refer to Genie Properties, page 62.

Genie Icons

Refer to Genie Icons, page 63.

Representation

The table describes the genies of the `gpl_equipmentmodule` library:

Graphic symbol	Genie name	Description
	*PanelState*	The genie Indicates the below statuses. The symbol displays: • The current equipment state. • The current state of the sequence. • The step being executed. • The strategy that is applied (if configured and selected). • The time elapsed for the current step.
	PanelWithButtons	The genie Indicates the below statuses. The symbol displays: • The current equipment state. • The current state of the sequence. • The step being executed. • The strategy that is applied (if configured and selected). • The time elapsed for the current step. • Control buttons (START, HOLD, PAUSE, RESUME, STOP, ABORT, RESET).
	*PanelWithButtons_2data*	The genie Indicates the below statuses. The symbol displays: • The current equipment state. • The current state of the sequence. • The step being executed. • The strategy that is applied (if configured and selected). • The time elapsed for the current step. • Control buttons (START, HOLD, PAUSE, RESUME, STOP, ABORT, RESET). • Two configurable data (Data1 and Data2).
	*PanelWithButtons_4data*	The genie Indicates the below statuses. The symbol displays: • The current equipment state. • The current state of the sequence. • The step being executed. • The strategy that is applied (if configured and selected). • The time elapsed for the current step. • Control buttons (START, HOLD, PAUSE, RESUME, STOP, ABORT, RESET). • Four configurable data (data1, data2, data3, and data4).

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

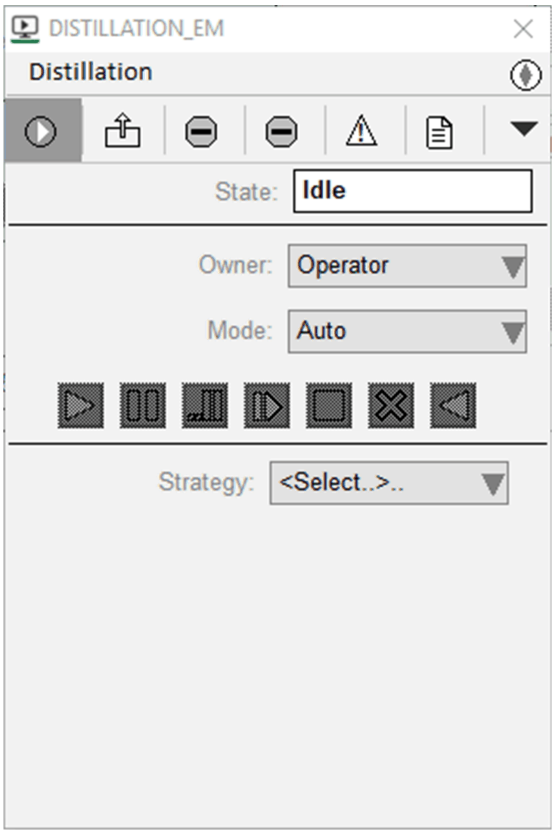
Available Tabs

During operation, clicking an equipment module control genie opens a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Output Parameter
  - State Machine
  - Alarm, page 67
  - Event, page 69
- Tabs for optional functions, which appear only if configured:
  - InitCond, page 867
  - Failure, page 78

Operator Tab

The figure shows an operator tab:

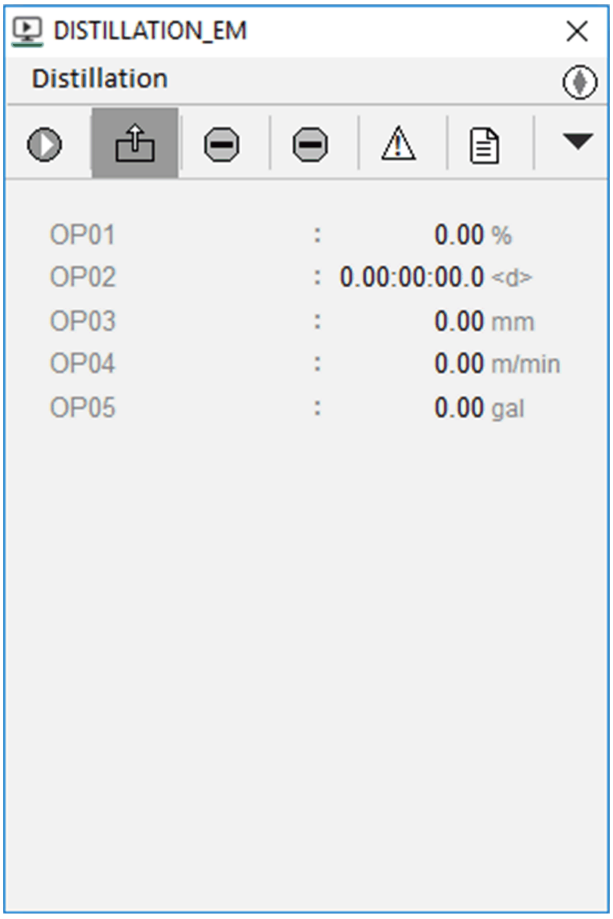


The table describes the control function of an operator tab:

Lable	Item name	Description	Security parameter, page 853
State	State	State	–
Owner	OwnerSelect	Current owner of equipment module control. For more details, refer to operating owner, page 58.	Set owner
Mode	Mode	Allows you to view and change the mode	Set auto mode, Set semi mode, set manual
Strategy	Strategy, OwnerSelect	Strategy of equipment module	Set strategy

Output ParameterTab

The figure shows an output parameter tab:

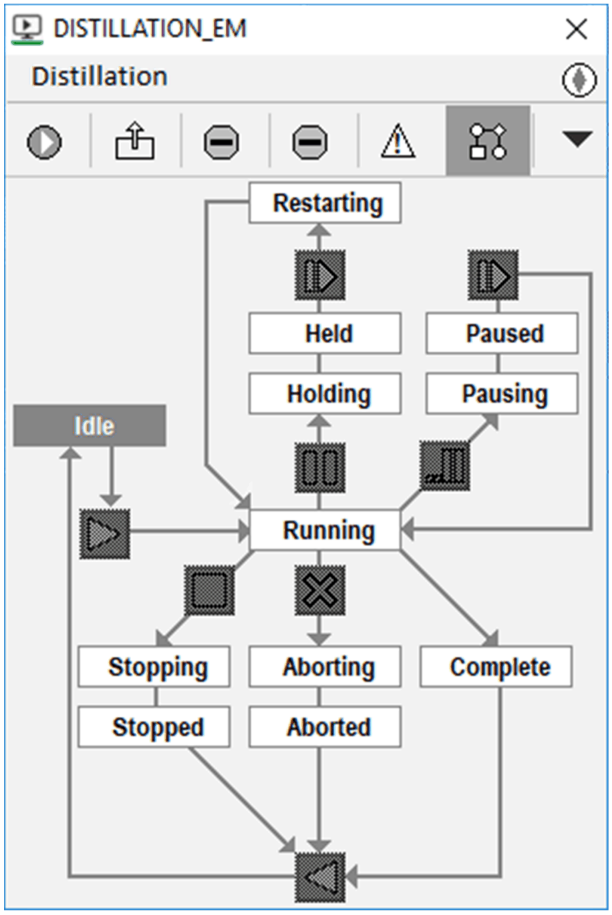


The table describes the control function of an output parameter tab:

Lable	Item name	Description
OP{x}	OP{x}	Output parameter {x}
NOTE: {x} represents from 01 to 05.		

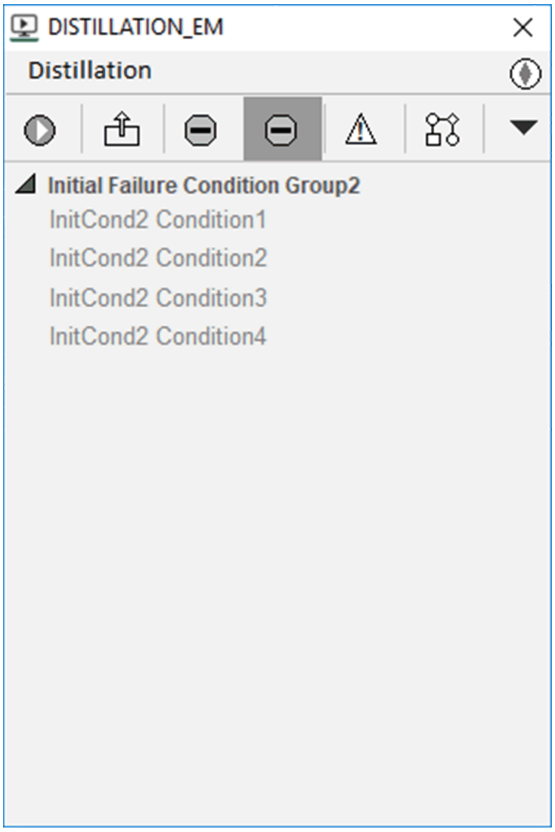
State Machine Tab

The figure shows a state machine tab:



InitCond Tab

The figure shows an InitCond tab:



Items

Overview

This section describes the variables, equipstate variable, abnormal and action required conditions, alarms, and events items of equipment module.

Variables

The table describes the variable items that are used by supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`Transition{x}`	First transition status	Done/ Pending	*{y}*
NOTE: {x} represents the group, range from 01 to 06. {y} represents the condition, range from 0 to 6.			
`AutoResetDisable`	Auto reset	Disabled/Not Disabled	*Bit15*
Bit items derived from ConfigurationWord.			
`OwnerSelect`	Owner of equipment module	Operator/ Program	*Bit0*
`SemiAutomatic`	Semi automatic mode command	SemiAuto-matic/Not Semi Automatic	*Bit1*

Item name	Description	Enumeration	Address
Manual	Manual mode command	Auto/Manual	*Bit2*
NextStep	Next step command	On/Off	*Bit3*
ScrollUp	Scroll up command		*Bit4*
ScrollDown	Scroll down command		*Bit5*
Bit items derived from state.			
Idle	Idle state of EM	–	*Bit0*
Running	Running state of EM		*Bit1*
Held	Held state of EM		*Bit2*
HeldForError	Status reached after the pause sequence is completed and the next steady status is reached to a detected failure		*Bit3*
Stopped	The stop sequence is completed and requires an initialization command to go to Idle status		*Bit4*
Paused	Pause sequence is completed and the next steady status is reached		*Bit5*
Aborted	An aborting sequence has been completed and requires an initialization command reset to go to Idle status		*Bit6*
Complete	Normal operation has finished and requires an initialization order reset to return to the Idle status		*Bit7*
Holding	A hold command is received and the sequence that allows the process to be changed to a known status is executed after this sequence is finished it automatically goes to the held status		*Bit8*
HoldingForError	A detected failure is automatically detected in the process for example a detected failure in a equipment module and the sequence that allows the process to be to a known status is executed and after this sequence is finished it automatically goes to stop state.		*Bit9*
Restarting	Restart command is received while the sequence is being held and the sequence runs to return to the normal operation sequence running and the normal operation status is set automatically after the sequence is complete		*Bit10*
Pausing	A pause command is received for a short period while the normal operation was being executed and the normal sequence runs until it automatically goes to paused status		*Bit11*
Stopping	An order to stop has been received and the sequence to finish the sequence in a controlled way is executed and stopped operation status is automatically activated after the sequence is complete		*Bit12*
Aborting	An order to end abort has been received a fast stop which is not necessarily controlled is executed and aborted operation status is automatically activated after the sequence is completed		*Bit13*
Bit items derived from Disstrategy			
Disstrategy{x}	Disabling the execution of strategy {y}	–	*Bit{z}*
NOTE: {x} represents the group, range from 01 to 16. {y} represents the condition, range from 1 to 15. {z} represents the condition, range from 1 to 16.			
Bit items derived from command.			
Start	Allows the normal operation of the sequence running to be started	–	*Bit0*
Hold	Allows the sequence to be stopped and the stopping sequence to be run		*Bit1*
Restart	Allows the sequence for continuing the operation and resume normal operation running		*Bit2*
Stop	Allows the sequence like running or pausing or paused or holding or held or restarting to be stopped and the stopping sequence to be run		*Bit3*
Reset	Triggers the transition to idle status in complete or aborted or stopped status		*Bit4*
Pause	Allows normal execution to be paused in the next steady sequence status		*Bit5*

Item name	Description	Enumeration	Address
Resume	Allows normal operation of the sequence to be resumed from the paused status		*Bit6*
Abort	Allows the sequence to be aborted in any status except idle or Complete or aborting or aborted and the aborting sequence to be run		*Bit7*
IO Items			
StatusWord	Status word	–	EMCTLGP_ST.STW
ConfigurationWord	Configuration word		EMCTLGP_ST.CFGW
State	Disable Strategy		EMCTLGP_ST.STATE
Disstrategy	State		EMCTLGP_ST.STATE
Command	command		EMCTLGP_ST.COMMAND
Calculated Items			
EquipState	Output information For more details, refer to the topic describing the items of equipstate calculated variable, page 870.	–	GPL_EMEquipState
Owner	Current owner of the equipment module For more details, refer to the topic describing the items of calculated variable, page 59.	–	GPL_OwnerBasic
Mode	Current mode of the equipment module pattern For more details, refer to the topic describing the items of calculated variable, page 59.	–	GPL_SeqEMMode
Abnormal	Shows abnormal condition of the equipment module pattern For more details, refer to the topic describing the items of abnormal conditions, page 869.	–	GPL_SeqEMAbnormal
ActionRequired	Shows action required condition of the motor For more details, refer to the topic describing the items of action required conditions, page 870.		SeqEMAction-Required

Abnormal Conditions

The table describes the abnormal conditions of equipment module pattern:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
AbnormalOwner	Abnormal Owner status appears, if the owner is not program.	*Bit0*
AbnormalMode	Abnormal mode status appears, if the Mode is not normal.	*Bit1*
AbnormalOperatorTab	Abnormal status of operator tab. This condition appears, if any of alarm is not enabled.	*Bit8*
AbnormalFailTab	Abnormal status of fail tab. This condition appears, if any of alarm is not enabled.	*Bit10*
AbnormalInterlockTab	Abnormal status of interlock tab. This condition appears, if any of alarm is not enabled.	*Bit11*

Item name	Description	Address
AbnormalSequenceTab	Abnormal status of sequence tab. This condition appears if the Mode is not normal.	*Bit13*
AbnormalInitCondTab	Abnormal status of InitCond tab. This condition appears if initial detected failure condition is active.	*Bit15*

Action Required Conditions

The table describes the action required conditions of motor:

Item name	Description	Address
Bit items derived from ActionRequired variable word.		
ActionOperatorTab	Action required status of operator tab.	*Bit8*
ActionFailTab	Action required status of fail tab.	*Bit10*
ActionInterlockTab	Action required status of interlock tab.	*Bit11*
ActionSequenceTab	Action required status of sequence tab.	*Bit13*
ActionInitCondTab	Action required status of InitCond tab.	*Bit15*

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	Conditions	Enumeration
Active	iRunning	Running
Passive	Idle x Complete x Paused x Held x HeldForError x Stopped x Aborted	Stopped
Transitioning Passive	Restarting	Held
Transitioning Active	Pausing x Holding x HoldingForError x Stopping x Aborting	Holding For Error

Alarms

The table describes the advanced alarm items that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the state:

Item name	Description	Expressions
HeldForError	Held for error	GPL_DualBitCheck (State, 3,9,1)

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
SateAlarm	All changes in the equipment module pattern status and output.
ModeAlarm	All changes in the mode and owner.
ActionAlarm	All actions by the operator.

EMCTLGP - Example of Use

What's in This Chapter

Description 871
DFB Representation 872
Reusable Equipment Module Logic 872

Overview

This chapter describes the example use of the EMCTLGP DFB.

Description

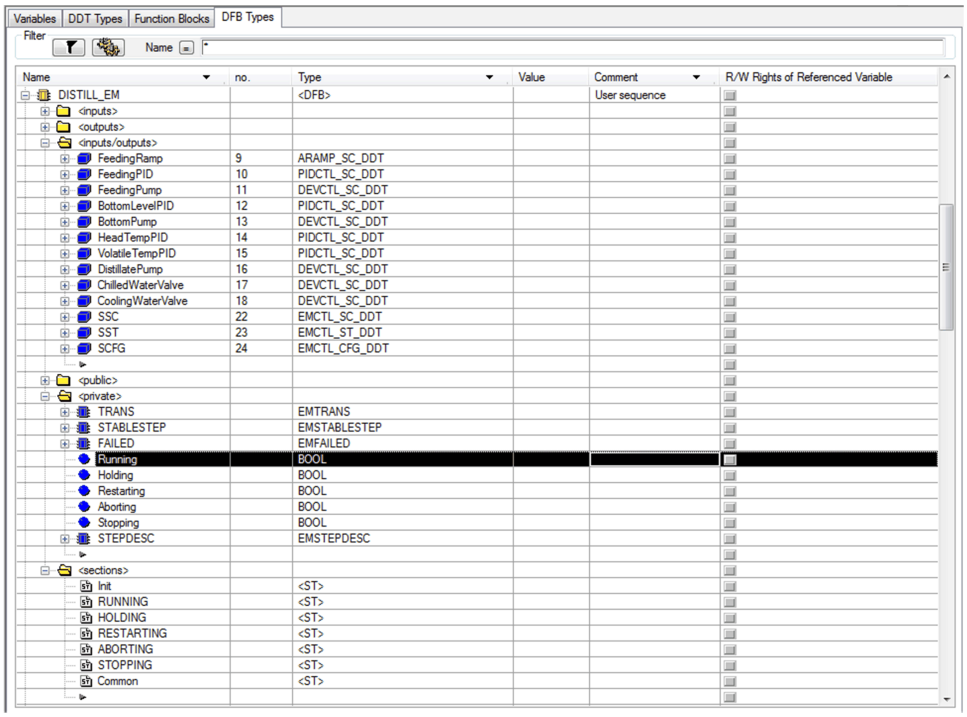
General

This chapter gives an example of an advanced equipment module controlled by the EMCTL, EMTRANSGP, EMSTABLESTEPGP, EMFAILEDGP, and EMSTEPDESC DFBs.

The objective of the sequence is to control the loading of water in a process unit. For this purpose, a reusable component (DFB) is created so that you can use this sequence in different process units by only changing the connections of the component pins.

Reusable Sequence Definition

The reusable DFB of the user, DISTILL_EM, declares the EMCTL EMCTL_ Instance variable to manage the state of the sequence.



Variables | DDT Types | Function Blocks | DFB Types

Filter — Name: *

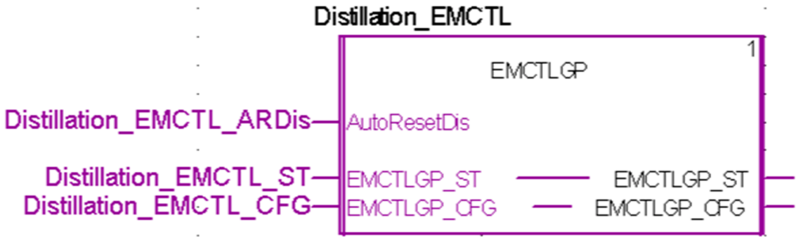
Name	no.	Type	Value	Comment	R/W Rights of Referenced Variable
DISTILL_EM		<DFB>		User sequence	
<inputs>					
<outputs>					
<inputs/outputs>					
FeedingRamp	9	ARAMP_SC_DDT			
FeedingPID	10	PIDCTL_SC_DDT			
FeedingPump	11	DEVCTL_SC_DDT			
BottomLevelPID	12	PIDCTL_SC_DDT			
BottomPump	13	DEVCTL_SC_DDT			
HeadTempPID	14	PIDCTL_SC_DDT			
VolatileTempPID	15	PIDCTL_SC_DDT			
DistillatePump	16	DEVCTL_SC_DDT			
ChilledWaterValve	17	DEVCTL_SC_DDT			
CoolingWaterValve	18	DEVCTL_SC_DDT			
SSC	22	EMCTL_SC_DDT			
SST	23	EMCTL_ST_DDT			
SCFG	24	EMCTL_CFG_DDT			
<public>					
<private>					
TRANS		EMTRANS			
STABLESTEP		EMSTABLESTEP			
FAILED		EMFAILED			
Running		BOOL			
Holding		BOOL			
Restarting		BOOL			
Aborting		BOOL			
Stopping		BOOL			
STEPDESC		EMSTEPDESC			
<sections>					
Init		<ST>			
RUNNING		<ST>			
HOLDING		<ST>			
RESTARTING		<ST>			
ABORTING		<ST>			
STOPPING		<ST>			
Common		<ST>			

NOTE: Declaration of the auxiliary components manages the transitions (EMTRANSGP), steady-step flags (EMSTABLESTEPGP), and notifications to trigger the holding subsequence (EMFAILEDGP).

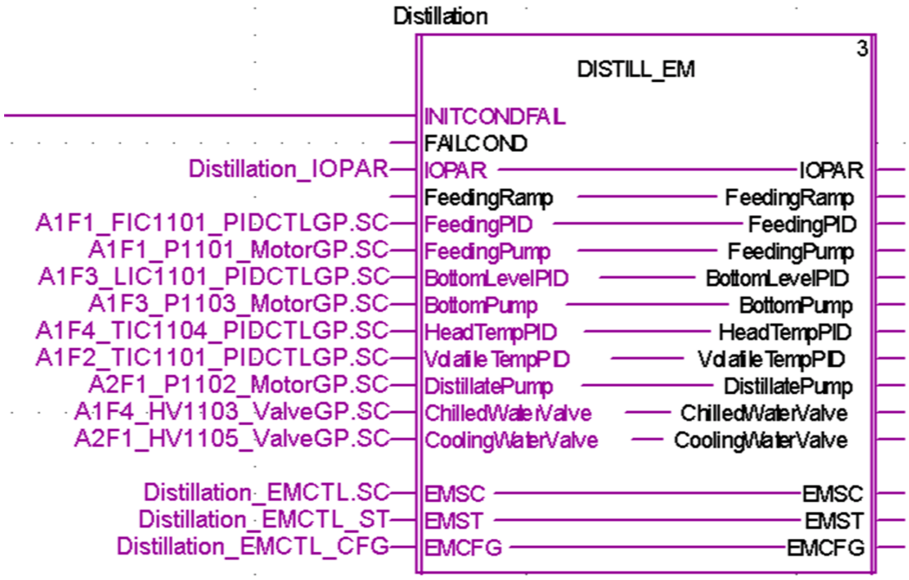
DFB Representation

Representation

The following diagrams include the calls to the sequence created by the user as well as to the control block that will control it (EMCTL).



NOTE: The call to the block controls the execution of the equipment module controller.



NOTE: The call to the functional block implements the control sequence of the user and will reuse for different process units if needed.

Reusable Equipment Module Logic

Equipment Module Logic With Description

The basic structure of the sequence as well as the steps that are defined for the sequence objective are included: Distillation. The parts of the code that make up the basic structure of the sequence are marked in bold.

(Initialization)

The following common actions have to be executed before the sequence execution:

```
Running := SSC.RUNNINGX;

Holding := SSC.HOLDINGX;

Restarting := SSC.RESTARTINGX;
```

```
Aborting := SSC.ABORTINGX;
Stopping := SSC.STOPPINGX;
```

NOTE: The default actions to be taken by DFB are defined in this section. However, if any section has to be added by user and actions need to be taken before execution of that section, it has to be added here.

(Running Subsequence)

In the running sequence, different strategies are implemented such as Start distillation, Stop distillation, Quick Stop.

In the running sequence, Distillation process started step by step in sequence.

```
Case SSC.Strategy of
```

1: (Start Distillation Strategy)

```
STEPDESC (1, 'Initialization', SSC, SST, SCFG);
STEPDESC (10, 'Refrigeration ON', SSC, SST, SCFG);
STEPDESC (20, 'Feeding ON', SSC, SST, SCFG);
STEPDESC (30, 'Distilling', SSC, SST, SCFG);
```

(* Blocking from restarting of the strategy *)

```
if SST.EMSTATE = 5 then
SSC.CSTEP := 30;
End_if;
```

NOTE: Definition of the descriptions for the subsequence steps.

(Steps and Transitions)

```
case SSC.CSTEP of
```

1: (Initial Step)

NOTE: First step with a value of 1. The allowed values for steps are 1 to 999.

(Check initial conditions)

```
If INITCONDFAIL then
FAILED (SSC);
```
(Initial conditions not satisfied)

```
end_if;
```

NOTE: Report detected failure to trigger the holding subsequence.

```
if (SST.EMSTATE = 16) then
TRANS (true, 'Jump to Step 20', 20, SSC, SST, SCFG);
```
(* Transition to Step 20 *)

```
elsif (SST.EMSTATE = 17) then
TRANS (true, 'Jump to Step 30', 30, SSC, SST, SCFG);
```
(* Transition to Step 30 *)

```
else
TRANS (true, 'Jump to Step 10', 10, SSC, SST, SCFG);
```
(* Transition to Step 10 *)

```
end_if;
```

NOTE: Unconditional transition to the next step.

10: (Refrigeration ON)

```
if SSC.ONENTRY then
```

NOTE: Execute the actions only for the first time.

```
ChilledWaterValve.REM := false;
```

```
ChilledWaterValve.LSP := true;

CoolingWaterValve.REM := false;

CoolingWaterValve.LSP := true;

end_if;

TRANS (ChilledWaterValve.HIGHPOS and CoolingWaterValve.
HIGHPOS and SSC.ETIME >= 30, 'Valves open', 20, SSC, SST, SCFG;
```

NOTE: The evaluation priority for these transitions is the order in which they are shown (up down; these transitions are not functional, and have only been included as an example in this case). You can define a maximum of 6 transitions.

20: (Feed ON)

```
if SSC.ONENTRY then

FeedingRamp.REM := false;

FeedingRamp.LTARGETSP := FeedRampSP;

FeedingRamp.LSTART := true;

FeedingPID.AUTO := true;

FeedingPump.REM := false;

FeedingPump.LSP := true;

BottomLevelPID.REM := false;

BottomLevelPID.LSP := BottomLevelPIDSP;

BottomLevelPID.AUTO := true;

BottomPump.LSP := true;

DistillatePump.REM := true;

end_if;

TRANS (BottomLevelPID.PV >= 50.0, 'Level >= 50%', 30, SSC, SST,
SCFG);)
```

30: (Distilling Started)

```
if SSC.ONENTRY then

HeadTempPID.REM := false;

HeadTempPID.LSP := HeadTempPIDSP;

HeadTempPID.AUTO := true;

VolatileTempPID.REM := false;

VolatileTempPID.LSP := VolatileTempPIDSP;

VolatileTempPID.AUTO := true;

end_if;

TRANS (True, 'Return to IDLE', -1, SSC, SST, SCFG);

end_case;
```

2: (Stop distillation Strategy)

```
STEPDESC (1, 'Initialization', SSC, SST, SCFG);

STEPDESC (10, 'Heating OFF', SSC, SST, SCFG);

STEPDESC (20, 'Discharge C11', SSC, SST, SCFG);

STEPDESC (30, 'Discharge D11', SSC, SST, SCFG);
```

```
STEPDESC (40, 'Close Valves', SSC, SST, SCFG);
(* Blocking from restarting of the strategy *)
if (SST.EMSTATE = 1 or SST.EMSTATE = 3) then
SSC.CSTEP := 40;
end_if;
(Steps and Transitions)
case SSC.CSTEP of
1: (* Initial Step. DO NOT REMOVE *)
(* Check initial conditions *)
IF INITCONDFAIL then (* To check the Initial Conditions *)
FAILED (SSC); (* Execute 'Holding for error' sequence *)
end_if;
TRANS (true, 'Jump to Step 10', 10, SSC, SST, SCFG); (* Transition to
Step 10 *)
10: (Heating OFF)
FeedingRamp.REM := false;
FeedingRamp.LTARGETSP := 0.0;
FeedingRamp.LSTART := true;
FeedingPID.MAN := true;
FeedingPID.LOP := 0.0;
FeedingPump.REM := false;
FeedingPump.LSP := false;
HeadTempPID.REM := false;
HeadTempPID.MAN := true;
HeadTempPID.LOP := 0.0;
VolatileTempPID.MAN := true;
VolatileTempPID.LOP := 0.0;
TRANS (SSC.ETIME >= 30, 'Wait 3 sec', 10, SSC, SST, SCFG);
20: (Discharge C11)
if SSC.ONENTRY then (* To be done only the first time the step is executed *)
BottomLevelPID.REM := false;
BottomLevelPID.MAN := true;
BottomLevelPID.LOP := 100.0;
BottomPump.REM := false;
BottomPump.LSP := true;
end_if;
TRANS (BottomLevelPID.PV <= 20.0 and SSC.ETIME >= 30, 'C11
Empty', 20, SSC, SST, SCFG);
if SSC.ONEXIT then (* To be done only when step exists *)
BottomLevelPID.REM := false;
```

```
BottomLevelPID.MAN := true;
BottomLevelPID.LOP := 0.0;
BottomPump.REM := false;
BottomPump.LSP := false;
end_if;
```

30: (Discharge D11)

```
IF SSC.ONENTRY THEN (* To be done only the first time the step is executed *
DistillatePump.REM := false;
DistillatePump.LSP := true;
end_if;
TRANS (DistillatePump.ILCKD and SSC.ETIME >= 30, 'D11 Empty',
30, SSC, SST, SCFG); (End of sequence)
if SSC.ONEXIT then (* To be done only when step exists *)
DistillatePump.REM := false;
DistillatePump.LSP := false;
end_if;
```

40: (* Close Water Valves *)

```
ChilledWaterValve.REM := false;
ChilledWaterValve.LSP := false;
CoolingWaterValve.REM := false;
CoolingWaterValve.LSP := false;
TRANS (((ChilledWaterValve.LOWPOS and CoolingWaterValve.
LOWPOS and SSC.ETIME >= 30) or (SST.EMSTATE = 1)), 'Valves
Closed', -1, SSC, SST, SCFG);
end_case;
```

3: (Quick Stop)

(* Step descriptions *)

(Steps and Transitions)

```
case SSC.CSTEP of
```

1: (* Transitions should not be included for emergency situations *)

```
FeedingRamp.REM := false;
FeedingRamp.LTARGETSP := 0.0;
FeedingRamp.LSTART := true;
FeedingPID.MAN := true;
FeedingPID.LOP := 0.0;
FeedingPump.REM := false;
FeedingPump.LSP := false;
BottomLevelPID.REM := false;
BottomLevelPID.MAN := true;
BottomLevelPID.LOP := 0.0;
BottomPump.REM := false;
```

```
BottomPump.LSP := false;
HeadTempPID.REM := false;
HeadTempPID.MAN := true;
HeadTempPID.LOP := 0.0;
VolatileTempPID.MAN := true;
VolatileTempPID.LOP := 0.0;
DistillatePump.REM := true;
DistillatePump.LSP := false;
ChilledWaterValve.REM := false;
ChilledWaterValve.LSP := false;
CoolingWaterValve.REM := false;
CoolingWaterValve.LSP := false;
TRANS (true, 'End', -1, SSC, SST, SCFG); (* To 'Aborted' state *)
(Stopping subsequence)
case SSC.CSTEP of
1:
TRANS (SSC.ETIME >= 10, 'End', -1, SSC, SST, SCFG); (* To 'Stopped' state after 5 seconds *)
End_case;
(Common Sequence)
(* Checks detected fail conditions *)
if FAILCOND then
FAILED (SSC); (* Forces to Hold the sequence*)
SST.EMSTATE := 1; (* Stopped *)
end_if;
(* Manual mode - based on the process step selected, EMSTATE is adapted *)
If SSC.Mode = 2 then
If SSC.Strategy = 1 then (* Start Distillation Strategy *)
if SSC.CSTEP = 1 then
SST.EMSTATE := 3; (* Ready *
elsif SSC.CSTEP = 10 then
SST.EMSTATE := 2; (* Starting *)
elsif SSC.CSTEP = 1=20 then
SST.EMSTATE := 16; (* Cooling ON *)
elsif SSC.CSTEP = 1=30 then
SST.EMSTATE := 17; (* Feeding ON *)
else;
end_if;
elsif SSC.Strategy = 2 then (* Stop Distillation Strategy *)
If SSC.CSTEP = 10 then
```

```
SST.EMSTATE := 3; (* Ready *)
else;
SST.EMSTATE := 10; (* Stopping *)
end_if;
else;
end_if;
end_if;
```

In the common sequence, logic for disabling the strategy, EMSTATE switching logic is based on the process states that are implemented.

```
Case SST.EMSTATE of 0: (* Off state *)
SST.DISSTRATEGY := 0; (* OFF state - Stop distillation strategy is disabled *)
SST.DISSTRATEGY02 := TRUE;
If SSC.Strategy <> 0 then
SST.DISSTRATEGY := 0;
SST.EMSTATE := 3; (* Ready *)
end_if;
1: (* Stopped *)
SST.DISSTRATEGY := 0; (* Stopped state - Stop distillation strategy is disabled
*)
SST.DISSTRATEGY02 := TRUE;
If (SSC.Strategy = 1 and not SST.HeldForError) then
SST.EMSTATE := 3; (* Ready *)
end_if;
2: (* Starting *)
SST.DISSTRATEGY := 0; (* Starting state - Start distillation strategy is disabled
*)
SST.DISSTRATEGY01 := TRUE;
SST.EMSTATE := 3; (* Ready *)
If (ChilledWaterValve.HIGHPOS and CoolingWaterValve.HIGHPOS
and SSC.ETIME >= 30) then
SST.EMSTATE := 16; (Feeding ON)
end_if;
3: (* Ready *)
If (not SST.HeldForError) then
If ChilledWaterValve.LSP = true and CoolingWaterValve.LSP = true
then
SST.EMSTATE := 2; (* Stopping *)
end_if;
SST.DISSTRATEGY02 := TRUE;
else
SST.EMSTATE := 1; (* Stopped *)
end_if;
```

5: (* Producing *)

```
If (FeedingRamp.LTARGETSP = 0.0 and (HeadTempPID.MAN = true and
HeadTempPID.LOP = 0.0)) then
```

```
SST.EMSTATE := 10; (* Stopping *)
```

```
end_if;
```

```
SST.DISSTRATEGY := 0; (* Producing state - Start distillation strategy is
disabled *)
```

```
SST.DISSTRATEGY01 := TRUE; (* Stopping *)
```

10: (* Stopping *)

```
If DistillatePump.LowPOS and FeedingPump.LOWPOS and
ChilledWaterValve.LOWPOS and CoolingWaterValve.LOWPOS and
BottomPump.LowPOS then
```

```
SST.EMSTATE := 1; (* Stopped *)
```

```
end_if;
```

```
If Not SST.Running then
```

```
SST.DISSTRATEGY := 0; (* Stopping state - Stop distillation strategy is disabled
*)
```

```
SST.DISSTRATEGY01 := TRUE;
```

```
end_if;
```

16: (* Cooling ON *)

```
If not (SSC.Strategy = 3) then
```

```
If FeedingPump.HIGHPOS and BottomPump.HIGHPOS then
```

```
SST.EMSTATE := 17; (* Feeding ON *)
```

```
end_if;
```

```
If Not SST.Running then
```

```
SST.DISSTRATEGY := 0; (* Cooling ON state - Stop distillation strategy is
disabled *)
```

```
SST.DISSTRATEGY02 := TRUE;
```

```
end_if;
```

```
else
```

```
If SST.Running then
```

```
SST.EMSTATE := 10; (* Stopping *)
```

```
end_if;
```

```
end_if;
```

17: (* Feeding ON *)

```
If not (SSC.Strategy = 3) then
```

```
If (HeadTempPID.LOP > 5.0 and VolatileTempPID.LOP >5.0 and
BottomLevelPID.PV >= 50.0) then
```

```
SST.EMSTATE := 5; (* Producing *)
```

```
end_if;
```

```
If Not SST.Running then
```

```
SST.DISSTRATEGY := 0; (* Feeding ON state - Stop distillation strategy is
disabled *)
```

```
SST.DISSTRATEGY02 := TRUE;
end_if;
else
If SST.Running then
SST.EMSTATE := 10; (* Stopping *)
end_if;
end_if;
end case;
```

EMLOGICGP - User Defined Process Logic

What's in This Chapter

Control 881

Overview

This chapter describes the components that provide the functionality of template, control services, and Supervision functions of the `EMLOGICGP` DFB.

Control

Overview

This section describes the `EMLOGICGP` DFB.

Description

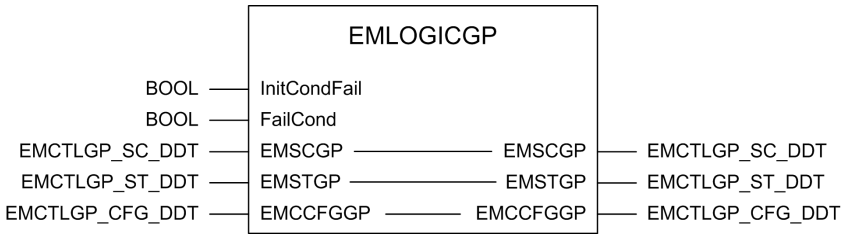
General

It is an unprotected function block in which user can add the logic to manage various equipment states, strategies and strategy execution states depending upon the process controlled by the user. Sample code is available in the function block for the user reference.

DFB Representation

Representation

The DFB that is used in the program has the following aspect at the section level. You can use it in any of the programming languages, although it is designed for use with the FBD language.



Inputs

Input Parameter Description

Parameter	Type	Description
InitCond-Fail	BOOL	The initial conditions should be satisfied for the equipment module operation. The result of the conditional summary block (CONDSUMGP) can be connected to this pin if user has multiple conditions.
FailCond	BOOL	The detected failure conditions during the operation of equipment module can be connected to this pin. The result of the conditional summary block (CONDSUMGP) can be connected to this pin if user has multiple conditions.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
EMSCGP_SC	EMCTLGP_SC_DDT, page 859	Corresponds to the SC public variable of the EMCTLGP block instance that is controlling the sequence execution.
EMSTGP_ST	EMCTLGP_ST_DDT, page 856	Provides the data necessary to communicate with the EMCTLGP block that is controlling the sequence. Refer to the EMCTLGP block, page 856 for more details.
EMCFGGP_CFG	EMCTLGP_CFG_DDT, page 859	Provides the data necessary to communicate with the EMCTLGP block that is controlling the sequence. Refer to the EMCTLGP block, page 856 for more details.

Auxiliary Functions of Equipment Module

What's in This Chapter

EMTRANSGP 883
EMFAILEDGP 885
EMSTABLESTEPGP 886
EMSTEPDESCGP 887

Overview

This section describes the function blocks that supplement the EMCTL DFB and that have the objective of facilitating the programming of the equipment module.

EMTRANSGP

Overview

This section describes the EMTRANSGP DFB.

Description

General

The main objective of EMTRANSGP DFB is to set up the control strategy sequence transitions based on the condition to be evaluated and the next step to be executed when this expression is fulfilled.

Additionally, the DFB manages the information necessary to graphically represent the state of the transitions and steps following the current step (DEST and DESTW fields of the EMCTL_ST_DDT equipment module state structure).

Include call to this function in the logic of the steps (normally in ST language) to program the transitions between the sequence steps.

Function Call

The function call is carried out from the steps of the control subsequences (Running, Restarting, Holding, and so on) normally in ST language to program the transitions between these steps.

If several EMTRANS calls are included in the same step and if more than one of them is fulfilled in the same step execution cycle, the step corresponding to the transition that is fulfilled and was programmed first will run. The number of EMTRANS calls from one single step is limited to 6.

```
TRANS (CONDITION := (*BOOL*),

DESC := (*string[22]*),

NEXTSTEP := (*INT*),

EMCTL_SC := (*EMCTL_SC_DDT*)

EMCTL_ST := (*EMCTL_ST_DDT*),

EMCTL_CFG := (*EMCTL_CFG_DDT*))
```

A function call example is included below:

4: (Wait for end of reactive dosing)

```
if SSC.ONENTRY then
BISEN2_REACT.REM:=false;
BISEN2_REACT.LSP:=true;
end_if;
BISEN2_REACT.LSPSEL:=(OP01 < (IP01 - 10.0)); (Speed 1 or 2)
TRANS ((IP01 - COLA_REACT) <= OP01, 'Dosing finished?', SSC.
CSTEP+1, SSC, SST, SCFG);
```

In the previous example, the TRANS call included in step 4 of the sequence corresponds to a `EMTRANSGP` block instance to which the following parameters are passed:

- `(IP01 - COLA_REACT) <= OP01OS`: Corresponds to the expression of the condition that needs to be fulfilled for the transition to be executed. You can include complex expressions that result in a Boolean value (true or false).
- `Dosing finished?`: Corresponds to the description of the transition.
- `SSC.CSTEP+1`: Corresponds to the next step that needs to be executed when the transition is fulfilled. In this case, the next step (step 5) is executed after the current step (step 4). You can include any expression jumping to an existing step in the sequence of the selected strategy.
- `SSC`: Corresponds to the `SC` public variable of the `EMCTL` block instance that is controlling the strategy sequence execution.
- `SST`: Corresponds to the `DDT_ST` variable of the `EMCTL` block instance that is controlling the strategy sequence execution.
- `SCFG`: Corresponds to the `DDT_CFG` variable of the `EMCTL` block instance that is controlling the strategy sequence execution.

Inputs

Input Parameter Description

Parameter	Type	Description
CONDITION	BOOL	1 = Indicates that the condition for executing the transition is fulfilled.
DESC	STRING[22]	Provides the description of the transition to be represented in the monitoring program. The character string is limited to 20 characters.
NEXTSTEP	INT	Subsequence step number to which a jump needs to be made when the transition is fulfilled. Refer to the CONDITION input. The permissible values for the steps go from 1 to 999 with the exception of NEXTSTEP to -1. This indicates that when the transition is fulfilled, the corresponding subsequence (to go, for example, from Running to Complete, from Holding to Held, and so on) will be completed.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
EMCTLGP_SC	EMCTLGP_SC_DDT	Corresponds to the SC public variable of the EMCTLGP block instance that is controlling the sequence execution.
EMCTLGP_ST	EMCTLGP_ST_DDT	Provides the data necessary to communicate with the EMCTLGP block that is controlling the equipment module. Refer to the EMCTLGP block, page 856 for more details.
EMCTLGP_CFG	EMCTLGP_CFG_DDT	Provides the data necessary to communicate with the EMCTLGP block that is controlling the equipment module. Refer to the EMCTLGP block, page 856 for more details.

EMFAILEDGP

Overview

This section describes the EMFAILEDGP DFB.

Description

General

The main objective of the EMFAILEDGP DFB is to report detected errors to the equipment module management block (EMCTL) during strategy sequence execution so that the *HoldingFE* subsequence is triggered.

Function Call

The function call is made from the Running, Restarting, Pausing, or Paused control subsequence steps normally in ST language with the purpose of reporting detected errors in the process that is being controlled. The detected error requires the hold sequence to be executed (*HoldingFE*).

A function call example is included below:

(Detected failure condition monitoring after initializations)

```
if EMCTL.CSTEP > 0 and INITCONDFAIL then
FAILED (EmCtl); (Trigger execution of the Hold sequence)
end_if;
```

In the previous example, the *FAILED* call that is included in the logic monitors the detected error conditions in each execution cycle (in this particular case, but could be associated with a specific step) corresponds to a EMFAILEDGP DFB block instance to which the following parameters are passed:

- Emctl: Corresponds to the name of the SC public variable of the EMCTL block instance that is controlling the execution of the sequence.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
EMCTLGP_SC	EMCTLGP_SC_DDT	Provides the data necessary to communicate with the EMCTLGP block that is controlling the sequence. Refer to the EMCTLGP block, page 856 for more details.

EMSTABLESTEPGP

Overview

The section describes the EMSTABLESTEPGP DFB.

Description

General

The main objective of the EMSTABLESTEPGP DFB is to report to the equipment module management block (EMCTL) that the current running sequence step corresponds to a step, that is, the re-entry point in the normal execution (Running) sequence executes after restarting execution (Restarting).

Function Call

The function call is performed from the Running control subsequence steps normally in ST language for the purpose of reporting. The current step is the entry point to which the program should return when the Running subsequence is resumed after finishing the Restarting subsequence. Basically, it enables to resume the sequence execution that verifies the appropriate actions are performed or not.

For example, suppose a valve has to be opened. After the valve is opened, the next step involves turning on a pump. While waiting for the valve to open, a detected failure that makes the sequence stop occurs (Holding subsequence execution). You can resume execution by running the restart (Restarting) subsequence. After this, the program returns to the normal running subsequence (Running). After the program goes back to this Running subsequence, the intention is to make another attempt at opening the valve (because most likely, it has been closed during the Holding subsequence) not to start the pump in proceeding directly. Therefore, in the valve opening step, the step is marked and does not call the function EMSTABLESTEPGP in the pump starting step.

The following is an example of a function call:

0: (End Water dosing)

```
if EmCtl.ONENTRY then
RAMP_AGUA.LSTART:=false;
FIC_AGUA.REM:=false;
FIC_AGUA.MAN:=true;
FIC_AGUA.LOP:=0.0;
VALV_AGUA.LSP:=false;
```

```
STABLESTEP (EmCtl1);
end_if;
TRANS (VALV_AGUA.LOWPOS, EMCTL.CSTEP+1, EmCtl);
```

In the previous example, the STABLESTEP call included in the logic that monitors detected failure conditions in each execution cycle (in this particular case, but it could be associated with a specific step) corresponds to a EMSTABLESTEPGP DFB block instance to which the following parameters are passed:

- Emctl: Corresponds to the name of the SC public variable of the EMCTL block instance that is controlling the sequence execution.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
EMCTLGP_SC	EMCTLGP_SC_DDT	Provides the data necessary to communicate with the EMCTLGP block that is controlling the sequence. Refer to the EMCTLGP block, page 856 for more details.

EMSTEPDESCGP

Overview

This section describes the EMSTEPDESCGP DFB.

Description

General

The main objective of the EMSTEPDESCGP DFB is to assign descriptions to the strategy execution steps that are defined in the strategy execution sequence to allow you to monitor them from the Supervision system.

Function Calls

The function call is made from the control subsequence steps (Running, Restarting, Holding, and so on) at the beginning of the code to provide the description of the subsequence steps for their representation in the monitoring program. It is normally written in ST language.

Include as many EMSTEPDESC calls as there are steps defined in the subsequence or as there are steps that need to be represented in the Supervision system.

The syntax of the function is the following:

```
STEPDESC (STEPNUMBER := (INT),
DESC := (string[18]),
EMCTL_SC := (EMCTL_SC_DDT),
EMCTL_ST := (EMCTL_ST_DDT),
EMCTL_CFG := (EMCTL_CFG_DDT),
```

The following is an example of a function call:

(Running or Pausing Subsequence (302 / 314))

(Step descriptions)

```
STEPDESC (1, 'Init', SSC, SST, SCFG);
STEPDESC (2, 'Valves ON', SSC, SST, SCFG);
STEPDESC (3, 'Wait Valves ON', SSC, SST, SCFG);
STEPDESC (4, 'Dosing Reactive', SSC, SST, SCFG);
STEPDESC (5, 'Motor OFF', SSC, SST, SCFG);
STEPDESC (6, 'Waiting for queue', SSC, SST, SCFG);
```

(Steps and Transitions)

```
case SSC.CSTEP of
```

1: (#RunningStep0# Initial Step)

(Check initial conditions)

```
if IC_FAIL then
FalloCI := true;
FAILED (SSC);
```
(Initial conditions not satisfied)

```
else
FalloCI := false;
```

(Initialization)

In the above example, the STEPDESC calls included at the beginning of the subsequence correspond to a `EMSTEPDESCGP` DFB block instance to which the following parameters are passed in case of the first call:

- 1: Corresponds to step number 1.
- `Init`: Corresponds to the definition of the step.
- `SSC`: Corresponds to the `SC` public variable of the `EMCTL` block instance that is controlling the sequence execution.
- `SST`: Corresponds to the `DDT_ST` variable of the `EMCTL` block instance that is controlling the sequence execution.
- `SCFG`: Corresponds to the `DDT_CFG` variable of the `EMCTL` block instance that is controlling the sequence execution.

Inputs

Input Parameter Description

Parameter	Type	Description
STEPNUMBER	INT	Number of the subsequence step. The values allowed for the steps are from 1 to 999.
DESC	STRING[22]	Description of the number of the subsequence step. The character string is limited to 17 characters. This is the value that will be shown in the sequence monitoring program.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
EMCTLGP_SC	EMCTLGP_SC_DDT	Provides the data necessary to communicate with the EMCTLGP block that is controlling the sequence. Refer to the EMCTLGP, page 856 block for more details.
EMCTLGP_ST	EMCTLGP_ST_DDT	Provides the data necessary to communicate with the EMCTLGP block that is controlling the sequence. Refer to the EMCTLGP, page 856 block for more details.
EMCTLGP_CFG	EMCTLGP_CFG_DDT	Provides the data necessary to communicate with the EMCTLGP block that is controlling the sequence. Refer to the EMCTLGP, page 856 block for more details.

Public Variables

Public Variable Description

Parameter	Type	Description
ORDERNUMBER	INT	Read-only public variable reserved for building the list of steps in manual mode that will be shown in the monitoring program.
MAXORDERNUMBER	INT	Read-only public variable reserved for building the list of steps in manual mode that will be shown in the monitoring program.

Phase Manager

What's in This Part

PHASEGP - Phase Manager 891
Auxiliary Functions of PHASEGP 930
PHPARxxGP 937
UserPhasexxGP 942
PHASEGP - Example of Use 944

Overview

This part provides a detailed description of the functions, pins, pin layout, and variables of the function blocks of the phase manager.

These function blocks do not reflect any specific installation.

⚠ WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate them.
- Review the implications of communication link interruptions and take actions to mitigate them.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, and applicable codes and regulations.
- Apply local accident prevention and safety regulations and guidelines.[1]
- Test each implementation of a system for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

PHASEGP - Phase Manager

What's in This Chapter

Template 891
Control 896
Supervision 906

Overview

This chapter provides a detailed description of the functions, pins, pin layout, and variables of the PHASEGP DFB.

Template

Overview

This section describes the functionality, parameters, and interfaces of the $PhaseGP template.

$PhaseGP - Phase Manager

General Description

The $PhaseGP process control module template is used to monitor and manage phases.

These templates do not reflect any specific installation.

⚠ WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate them.
- Review the implications of communication link interruptions and take actions to mitigate them.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, and applicable codes and regulations.
- Apply local accident prevention and safety regulations and guidelines.[1]
- Test each implementation of a system for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control*

and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

Modifying the Sections of the DFB Type Managing the Sub phases

To use phase control, you need to modify the sections of the DFB type that manages the sub phases (for example, *RUNNING*).

You can use the **UserPhasexxGP DFB Type** parameter, page 892 to create a user-defined DFB type.

NOTE: The entry has to be compliant with naming convention of Control Expert.

Parameters

Parameters

The `$PhaseGP` template defines operation of Phase control. Alternative sequence control input facets enable the modification of the sequence control that are available.

Control

The table describes the control parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Maximum phase parameter	`Enum`	Upto 5 (1)	Maximum number of phase parameters that the phase manages. Depending on the value that you select, the following services are activated in addition to the services that are not parameter-dependent: • No parameters: Only `UserPhase0Par` (Control). **NOTE:** With configurable buttons and data, and with control and data genies are not available if this selection is made. • Upto 5: ◦ `5Parameters` (Control) ◦ `UserPhase5Par` (Control) ◦ `Parameters1to5` (Supervision) • Upto 10: ◦ `10Parameters` (Control) ◦ `UserPhase10Par` (Control) ◦ `Parameters1to5` and `Parameters6to10` (Supervision) • Upto 16: ◦ `16Parameters` (Control) ◦ `UserPhase16Par` (Control) ◦ `Parameters1to5`, `Parameters6to10` and`Parameters11to16` (Supervision)
Configuration	Phase DFB name	`String`	–	Name of the DFB type that is generated by the `$UserPhasexxGP UL` facet, where xx is 00, 05, 10 or 16 depending on the value of **Maximum number of phase parameters**. If you leave the parameter blank, it generates a DFB type, which has the identifier of the instance as name. In any case, an instance of the DFB type is created in the section. **NOTE:** • In any case, a `$UserPhasexxGP UL` DFB type is created but not used (where xx is 00, 05, 10 or 16 depending on the **Maximum number of phase parameters**). • You have to verify that the parameter is configured prior to the generation of the respective instance.

Section	Parameter description	Type	Default	Additional remarks
	Disable auto reset	Boolean	False	Disable auto reset.
Initial conditions	For more details, refer to the topic describing the parameters of interlocks, page 81			
Failures	For more details, refer to the topic describing the parameters of detected failure			

Supervision

The table describes the supervision parameters:

Section	Parameter description	Type	Default	Additional remarks
Operation	Normal Owner	Enum	Program (1)	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Security	Set Owner	Enum	Operator (1)	Privilege to change owner. This parameter is used in Operator tab, page 913.
	Set reset		Operator (Confirmed) (10)	Privilege to reset. This parameter is used in Operator tab, page 913.
	Set Alarm Acknowledge		Operator (1)	Privilege to acknowledge alarm. This parameter is used in Alarm tab, page 67.
	Set bypass conditions		Supervisor (2)	Privilege to bypass conditions.
	Set semi mode		Operator (1)	Privilege to change set semi mode. This parameter is used in Operator tab, page 913.
	Set manual mode			Privilege to change set manual mode. This parameter is used in Operator tab, page 913.
	Set next step			Privilege to change set next step. This parameter is used in Operator tab, page 913.
	Set start			Privilege to change set start. This parameter is used in Operator tab, page 913.
	Set hold			Privilege to change set hold. This parameter is used in Operator tab, page 913.
Security	Set restart		Operator (1)	Privilege to set restart. This parameter is used in Operator tab, page 913.
	Set stop			Privilege to set stop. This parameter is used in Operator tab, page 913.
	Set reset			Privilege to set reset. This parameter is used in Operator tab, page 913.
	Set pause			Privilege to set pause. This parameter is used in Operator tab, page 913.
	Set resume			Privilege to set resume. This parameter is used in Operator tab, page 913.

Section	Parameter description	Type	Default	Additional remarks
	Set abort			Privilege to set abort. This parameter is used in Operator tab, page 913.
	Set strategy			Privilege to set strategy. This parameter is used in Operator tab, page 913.
	Set Control Button1			Privilege to set control button1. This parameter is used in Operator tab, page 913.
	Set Control Button2			Privilege to set control button2. This parameter is used in Operator tab, page 913.
Asset and Display	For more details, refer to the topic describing the parameters of asset and display, page 43.			

Data

The table describes the data parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Phase type	Enum	InBatch (0)	Select either InBatch or recipe manager.
	Phase tool communication with controller (0 - through OFS; 1 - through Citect) **NOTE:** This Parameter is only Supported in legacy $PhaseGP.	Boolean	False	–
	Item 1 parameter	String	–	Refer to items for list of items, page 923
	Item 2 parameter		–	
	Description for control button1		–	Disables the control button 1, if description is not configured. **NOTE:** This parameter is only applicable for InBatch.
	Description for control button2			Disables the control button 2, if description is not configured. **NOTE:** This parameter is only applicable for InBatch.
Alarm	Held for error alarm description	String	@(Held For Error)	–
	Held for error alarm group		Failure	–
	Held for error alarm severity	Enum	Low (4000)	–
Operation	Enable log	Enum	Ready & Running (8)	It allows to define the normal owner groups and any other owner who has logged in indicates an abnormality in graphics.
Disable Button	Start command	Boolean	False	Disables the normal operation of the phase (Running) to be started.
	Stop command			Disables the phaseGP to be stopped (Running, Pausing, Paused, Holding, Held, or Restarting) and the transition of state from running to stopping (Stopping).
	Hold command			Disables the phaseGP to be put on hold (Holding, HoldingForError)..
	Pause command			Disables normal execution (Running) to be paused in the next steady phase status.
	Restart command			Disables the phaseGP for continuing the operation (Restarting) and resume normal operation (Running).

Section	Parameter description	Type	Default	Additional remarks
	Abort command			Disables the phaseGP to be aborted (in any status except Idle, Complete, Aborting, and Aborted) and the aborting state achieved from running (Aborting).
	Reset command			Disable the triggering of the transition to idle status (in Complete, Aborted, or Stopped status).
Strategy	Strategy {x} name	String	–	–
	Strategy {x} parameters (i.e. 2,3,6)			–
NOTE: {x} represents the group, range from 1 to 8.				
Event	Enable log	Boolean	False	It allows to historize the operator actions.
Historize	Command	Boolean	False	It allows to historize the change in commands.
	State			It allows to historize the chang in state.
	Current step description			It allows to historize the change in current step description.
	New Step			It allows to historize the change in new step.
	Equipment State			It allows to historize the change in equipment state.
	Phase mode			It allows to historize the change in phase mode.
	Current Step			It allows to historize the change in current step.
Initial conditions	For more details, refer to the topic describing the parameters of interlocks, page 81			
Failures	For more details, refer to the topic describing the parameters of detected failure			
Input Parameter Configuration				
	Input parameter {x} Description	String	–	–
	Input parameter {x} format		####.##	–
NOTE: {x} represents the group, range from 01 to 16.				
Output Parameter Configuration				
	Output parameter {x} Description	String	–	–
	Output parameter {x} format		####.##	–
NOTE: {x} represents the group, range from 01 to 16.				
Historize				
	Input parameter {x}	Boolean	False	–
	Output parameter {x}	Boolean	False	–
NOTE: {x} represents the group, range from 01 to 16.				

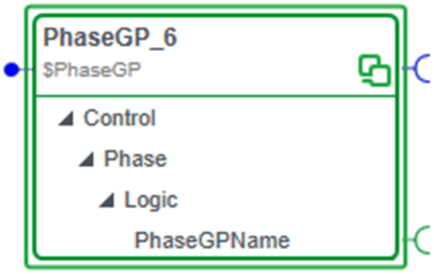
Hyperlink

For more details, refer to the topic describing the parameters of hyperlink services, page 50.

Interfaces

Interfaces

This figure shows the $PhaseGP (also applicable for $PhaseCE) template as it appears in Links Editor:



Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

Overview

This chapter describes the PHASEGP DFB.

Description of Phase Manager

Overview

The operation of the function block that is available for Phase Manager is described in this chapter.

The PHASEGP DFB belongs to the General Purpose Library and is used to monitor and manage control phases with tool. The module is based on the ISA-S88 Standard, Part 1 (S88.01i) and implements the status controller defined in this standard for controlling procedure model components (phases, operations, and so on). Prior knowledge about the tool will facilitate to understand the phase manager block.

Function Description

You can use this DFB to change states in the process. In compliance with ISA-S88.01i, you can use this DFB to initiate start, stop or emergency phases. In processes, this DFB can be used to set up phases and operations.

The DFB implements phase state management and processes the commands received from:

- The Supervision system (when phases are manually controlled by the operator)
- The control subsystem (in which case the subsystem sends the commands and checks the status of the phase to determine what action to take)
- Other control phases

The table compares the functions that are available in the DFB:

Function	PHASE
Management of state machine and commands according to ISA-S88	x
Management of operating modes according to ISA-S88: Manual (enables to change from one phase step to any other phase step), Semi (step advance with confirmation from the operator), and Auto	x
Owner Management: Operator/Program	x
Step time automatically calculated by the control block	x

Function	PHASE
Automatic management of transitions through the PHTRANSGP function block	x
Automatic management of phase detected failure through the PHFAILEDGP function block	x
Automatic management of phase steady-step marking through the PHSTABLESTEPGP function block	x
Possibility of displaying the phase graphically (active step and next level of transitions and steps) from the monitoring subsystem	x
NOTE: 'x' indicates that the functions are available in the DFB.	

Program the steps and transitions that the phases implement (through ST or SFC language) while the component (PHASEGP) allows the programmer to:

- Manage the statuses of the phase and
- Process the commands received from other levels as described above.

As a minimum, implement the subphase that implements normal operation (RUNNING).

Other user can implement subphases that attend to commands received from other levels of the control system or that react to detected failures in the process (HOLDING, RESTARTING, STOPPING, and ABORTING) detected by the phase itself.

The table describes the main functions of the DFB:

Function	Description
Processing States and Commands	The DFB processes the commands received and determines the status of the control phase with the objective of determining the subphases (normal, hold, continue, abort, so on.) and which step needs to run in each cycle of the program.
Managing Entry Points in the Phase	The DFB enables to retain the step number with which the execution of the normal phase (Running) needs to be restarted (Restart) after being restarted from held status (Held).
Handling of Detected Failures	The component manages controller detected failures (without user intervention from the Supervision system). The component manages this detected failure and triggers the execution of the detected failure handling subphase.
Auto reset disable	Automatic reset (*Stopped* →*Idle*, automatically performs reset operation without consuming any extra execution cycle) can be disabled by setting the input pin (*AutoResetDis*) to a logical high when Owner is Operator.

Definition of Phase Statuses, Phase Statuses, Phase Commands, and Phase Commands

Statuses

The table describes the phase statuses and their descriptions:

Status	Description
Idle	Inactive operation. Waiting for command to start (**Start**) running.
Running	Normal operation.
Held	Operation put on hold. Waiting for command to Restart (**Restart**) to get operation to running.
HeldForError	Operation put on hold due to detected fail condition. Waiting for command to Restart (**Restart**) to get operation to running.
Stopped	Operation stopped. Waiting for command to get state (**Reset**) to Idle.
Paused	Operation paused. Waiting for command to get state (**Resume**) to running.

Status	Description
Aborted	Operation canceled. Waiting for command to get state (**Reset**) to get operation to Idle.
Complete	Normal operation has finished. Requires an initialization order (**Reset**) to return to the Idle status.
Holding	A hold (**Hold**) command is received and the phase that allows the process to be changed to a known status is executed. After this phase is finished, it automatically goes to the held (Held) status.
HoldingForEr-ror	A detected failure is automatically detected in the process (for example, a detected failure in a control module), and the phase that allows the process to be changed to a known status is executed. After this phase is finished, it automatically goes to the held (HeldForError) status.
Restarting	A restart command (**Restart**) is received while the phase is being held (Held). The phase returns to the normal operation phase (Running). The normal operation status (Running) is set automatically after the phase is complete.
Pausing	A pause (**Pause**) command is received for a short period while the normal operation was being executed (Running). The normal phase runs until it automatically goes to paused (Paused) status.
Stopping	An order to stop (**Stop**) has been received. To finish the phase in a controlled way is executed (it is not possible to continue with it). The stopped operation status (Stopped) is automatically activated after the phase is complete.
Aborting	An order to end (**Abort**) has been received. A fast stop, which is not necessarily controlled is executed. The aborted operation status (Aborted) is automatically activated after the phase is completed.
Interlocked	Operation cannot start as initial conditions are not met.

The (Held) status can also be reached (without receiving a command) if the control phase itself detects a problem situation. The DFB has a signal that allows this situation to be communicated.

The DFB has the necessary means to be able to restart the Running phase from the last steady step in which the initial run was held by a **Hold** command.

Phase Statuses

The table describes the phase statuses and their descriptions:

STATE	Description
PhReady	Ready for operation. The phase state is in Idle.
PhRun	Normal operation. The phase can be in Running, Restarting, Stopping, Aborting, Holding, Pausing and Paused states.
PhHeld	Operation put on hold. The phase state is in Held.
PhAborted	Operation canceled. The phase can be in Stopped or Aborted states.
PhDone	Normal operation has finished. The phase state is in Completed.
PhInterlocked	Operation cannot start from tool as initial conditions are not met or Owner is Operator. The phase state is in Interlocked.

Commands

The table describes the phase commands and their descriptions:

Status	Description
Start	Allows the normal operation of the phase (Running) to be started.
Hold	Allows the phase state to be put on hold (Holding, HoldingForError).
Restart	Allows the phase state for continuing the operation (Restarting) and resume normal operation (Running).

Status	Description
Stop	Allows the phase state to be stopped (Running, Pausing, Paused, Holding, Held, or Restarting) and the transition of state from running to stopping (Stopping).
Reset	Triggers the transition to idle status (in Complete, Aborted, or Stopped status).
Pause	Allows normal execution (Running) to be paused in the next steady phase status.
Resume	Allows normal operation of the phase (Running) to be resumed from the paused (Paused) status.
Abort	Allows the phase state to be aborted (in any status except Idle, Complete, Aborting, and Aborted) and the aborting state achieved from running (Aborting).
Comple-ted	Allows phase state to reach quiescent states or final state from certain transient states.
Stable-step	1 = Sends a command to mark the current step as a steady step, that is, the step with which the phase will resume after its is held or the step on which the phase will be held after the phase is paused. 0 = The signal is automatically set to 0 by the DFB after the signal is processed.

Phase Commands

The table describes the phase commands and their descriptions:

PHCOMMAND	Description
PhStart	Allows the normal operation of the phase (*PhRun*) to be started.
PhHold	Allows the phase to be put on hold (*PhHeld*).
PhRestart	Allows the phase for continuing the operation and resume normal operation (*PhRun*).
PhReset	Triggers the transition to Ready status *PhReady* when phase state is in *PhDone* or *PhAborted*. The command is automatically triggered by phase tool when *PhState* is Done or Aborted.
PhAbort	Allows the phase to be aborted (in any phase status except Ready, Done, Run (Aborting phase state) and Aborted).

Status Matrix

Description

The following table describes the phase status controller that the DFB implements:

Initial State	Command	Activity	Final State (without command)
Idle	Start	Running	–
Running	Stop	Stopping	Stopped
	Hold	Holding	Held
	Pause	Pausing	Paused
	Abort	Aborting	Aborted
Complete	Reset	Idle	–
Pausing	Stop	Stopping	Stopped
	Hold	Holding	Held
	Abort	Aborting	Aborted
Paused	Stop	Stopping	Stopped
	Resume	-	Running

Initial State	Command	Activity	Final State (without command)
	Abort	Aborting	Aborted
	Hold	Holding	Held
Holding	Stop	Stopping	Stopped
	Abort	Aborting	Aborted
Held	Stop	Stopping	Stopped
	Restart	Restarting	Running
	Abort	Aborting	Aborted
Restarting	Stop	Stopping	Stopped
	Hold	Holding	Held
	Abort	Aborting	Aborted
Stopping	Abort	Aborting	Aborted
Stopped	Reset	Idle	–
	Abort	Aborting	Aborted
Aborting	No action possible		
Aborted	Reset	Idle	–

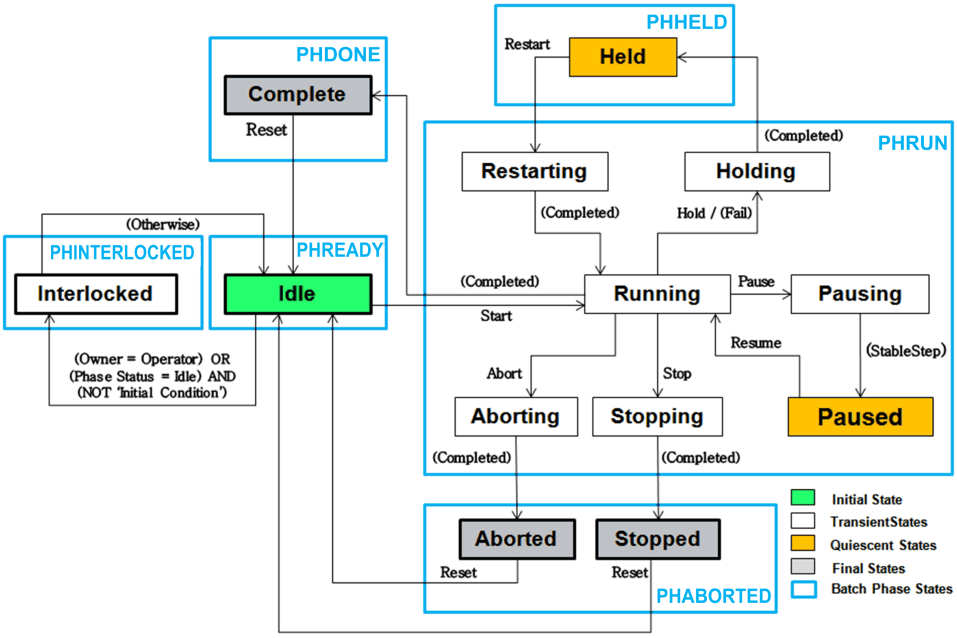
The Held status can also be reached (without receiving a command) if the control phase itself encounters a detected error. The block has a signal that allows this situation to be communicated.

The block has the necessary means to be able to restart the **Running** phase from the last steady step in which the initial run was held by a **Hold** command.

Status Controller

Description

The following diagram describes the possible phases statuses and phase commands that the Control Expert component manages according to the ISA-S88 standard:



Next to the transition arrows, the phase commands that the component receives (from the Supervision system or another higher level phase) are shown in black,

and status/commands that phase obtains from other blocks (for example, user logic) are shown in parenthesis.

Operating Modes

Mode Table

The following table provides the description of the phase operating modes of PHASEGP DFB:

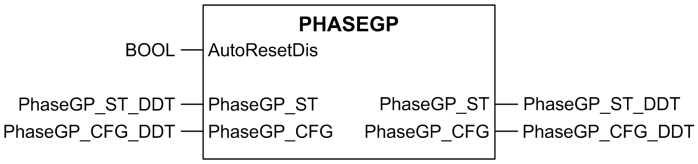
Operating mode	Implemented as:
Automatic	It is the normal running mode. The phase runs automatically based on the transitions defined in the controller.
Semiautomatic	In this operating mode, after the conditions defined in the transitions are met, you need to confirm the move to the next step.
Manual	The phase does not automatically advance to a step; user can manually change from one phase step to another.Once the phase step is selected, operating mode will change to semi-automatic and starts the phase step selected in manual mode.

DFB Representation

Representation

The DFB that is used in the program has the following aspect at the section level when imported.You can use it in any of the programming languages, although it is designed for use with the FBD language.

The figure shows an example of the PHASEGP DFB instance:



Implement the phase logic that controls the component in ST language to take advantage of the phase control potentials making the phase reusable. That is, you can run simultaneously several instances of the phase that you have implemented. This reduces engineering and maintenance efforts when it is necessary to set up the same control phases in several process units.

Inputs

Input Parameter Description

Parameter	Type	Description
AutoResetDis	BOOL	Disabling the automatic reset of the phase state, when Owner is Operator.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
PhaseGP_ST	PhaseGP_ST_DDT	Provides the necessary data for monitoring and controlling phase execution.
PhaseGP_CFG	PhaseGP_CFG_DDT	Provides the necessary data for background monitoring.

PhaseGP_ST_DDT Type

Name	Bit Rank Name (if any)	Bit Number (if any)	Type	Description
ETime	-		UDI-NT	The DFB calculates the execution time elapsed in tenths of a second for the current step of the control phase. If execution of the program is stopped and resumed, the accumulated execution time is considered but not the time that the program was stopped
PhState	-		UINT	Provides the status of the phase
	PhReady	0	BOOL	Inactive operation. Waiting for command to start (Start) running.
	PhRun	1	BOOL	Normal operation.
	PhHeld	2	BOOL	Operation put on hold. Waiting for command to Restart (Restart) to get operation to running.
	PhHeldForError	3	BOOL	Operation put on hold due to detected fail condition. Waiting for command to Restart (Restart) to get operation to running.
	PhStopped	4	BOOL	Operation stopped. Waiting for command to get state (Reset) to idle.
	PhPaused	5	BOOL	Operation paused. Waiting for command to get state (Resume) to running.
	PhAborted	6	BOOL	Operation canceled. Waiting for command to get state (Reset) to get operation to Idle
	PhHolding	8	BOOL	A hold (Hold) command is received and the phase that allows the process to be changed to a known status is executed. After this sequence is finished, it automatically goes to the held (Held) status.
	PhHoldingForEr-ror	9	BOOL	A detected failure is automatically detected in the process (for example, a detected failure in a control module), and the phase that allows the process to be to a known status is executed. After this sequence is finished, it automatically goes to the held (Heldforerror) status.
	PhRestarting	10	BOOL	Restart command (Restart) is received while the phase is being held (Held). The phase runs to return to the normal operation sequence (Running). The normal operation status (Running) is set automatically after the phase is complete.
PhState	PhPausing	11	BOOL	A pause (Pause) command is received for a short period while the normal operation was being executed

Name	Bit Rank Name (if any)	Bit Number (if any)	Type	Description
				(Running). The normal phase runs until it automatically goes to paused (Paused) status.
	PhStopping	12	BOOL	An order to stop (Stop) has been received. The phase to finish the order in a controlled way is executed (it is not possible to continue with it). The stopped operation status (Stopped) is automatically activated after the phase is complete.
	PhAborting	13	BOOL	An order to end (Abort) has been received. A fast stop, which is not necessarily controlled is executed. The aborted operation status (Aborted) is automatically activated after the phase is completed.
	PhDone	14	BOOL	Operation completed. Waiting for command to get state (Reset) to get operation to Idle.
	PhInterlocked	15	BOOL	Operation cannot start from phase tool as initial conditions are not met. or Owner is Operator.
PhCom-mand	-		UINT	Enables commands to be sent to the phase from the phase. The DFB sets the command to 0 after processed (successfully or not).
	PhStart	0	BOOL	Allows the normal operation of the phase (Running) to be started.
	PhHold	1	BOOL	Allows the phase to be put on hold (Holding, HoldingForError).
	PhRestart	2	BOOL	Allows the phase for continuing the operation (Restarting) and resume normal operation (Running).
	PhStop	3	BOOL	Allows the phase to be stopped (Running, Pausing, Paused,Holding, Held, or Restarting) and the transition of state from running to stopping (Stopping).
	PhReset	4	BOOL	Triggers the transition to idle status (in Complete, Aborted, or Stopped status).
	PhPause	5	BOOL	Allows normal execution (Running) to be paused in the next steady Sequence status.
	PhResume	6	BOOL	Allows normal execution (Running) to be paused in the next steady Sequence status.
	PhAbort	7	BOOL	Allows the phase for continuing the operation (Restarting) and resume normal operation (Running).
State	-		UINT	Provides the current status of the Phase.
	Idle	0	BOOL	Inactive operation. Waiting for command to start (Start) running.
	Running	1	BOOL	Normal operation.
	Held	2	BOOL	Operation put on hold. Waiting for command to Restart (Restart) to get operation to running.
	HeldForError	3	BOOL	Operation put on hold due to detected fail condition. Waiting for command to Restart (Restart) to get operation to running.
	Stopped	4	BOOL	Operation stopped. Waiting for command to get state (Reset) to Idle

Name	Bit Rank Name (if any)	Bit Number (if any)	Type	Description
	Paused	5	BOOL	Operation paused. Waiting for command to get state (Resume) to running
	Aborted	6	BOOL	Operation canceled. Waiting for command to get state (Reset) to get operation to idle
	Complete	7	BOOL	Normal operation has finished. Requires an initialization order (Reset) to return to the Idle status.
	Holding	8	BOOL	A hold (Hold) command is received and the phase that allows the process to be changed to a known status is executed. After this phase is finished, it automatically goes to the held (Held) status.
	HoldingForEr-ror	9	BOOL	A detected failure is automatically detected in the process (for example, a detected failure in a control module), and the phase that allows the process to be to a known status is executed. After this phase is finished, it automatically goes to the held (HeldForError) status.
	Restarting	10	BOOL	Restart command (Restart) is received while the phase is being held (Held). The phase returns to the normal operation phase (Running).
State	Pausing	11	BOOL	Operation cannot start from phase tool as initial conditions are not met.
	Stopping	12	BOOL	An order to stop (Stop) has been received. The phase to finish the order in a controlled way is executed (it is not possible to continue with it). The stopped operation status (Stopped) is automatically activated after the (Stopping) sub phase is completed.
	Aborting	13	BOOL	An order to end (Abort) has been received. A fast stop, which is not necessarily controlled is executed. The aborted operation status (Aborted) is automatically activated after the (Aborting) sub phase is completed.
	Interlocked	15	BOOL	Operation cannot start as initial conditions are not met.
Command	-		UINT	Phase commands from HMI
	Start	0	BOOL	Allows the normal operation of the phase (Running) to be started.
	Hold	1	BOOL	Allows the phase to be put on hold (Holding, Holdingforerror).
	Restart	2	BOOL	Allows the phase for continuing the operation (Restarting) and resume normal operation (Running).
	Stop	3	BOOL	Allows the phase to be stopped (Running, Pausing, Paused, Holding, Held, or Restarting) and the transition of state from running to stopping (Stopping).
	Reset	4	BOOL	Triggers the transition to idle status (in Complete, Aborted, or Stopped status).
	Pause	5	BOOL	Allows normal execution (Running) to be paused in the next steady phase status.
	Resume	6	BOOL	Allows normal operation of the phase (Running) to be resumed from the paused (Paused) status.
	Abort	7	BOOL	Allows the phase to be aborted (in any status except Idle, Complete, Aborting,

Name	Bit Rank Name (if any)	Bit Number (if any)	Type	Description
				and Aborted) and the aborting state achieved from running (Aborting).
STW	-		WORD	Phase status word. In this word, the *PHTRANSGP* auxiliary DFB is used to load the status of the transitions (maximum of six) from the current step of the phase to the steps in the next level of the phase, where 0 means that the conditions in the transition are not fulfilled and 1 means they are fulfilled.
	Status1Transition	0	BOOL	Status of the first transition
	Status2Transition	1	BOOL	Status of the second transition
	Status3Transition	2	BOOL	Status of the third transition
	Status4Transition	3	BOOL	Status of the fourth transition
	Status5Transition	4	BOOL	Status of the fifth transition
	Status6Transition	5	BOOL	Status of the sixth transition
	Fail	13	BOOL	Enables when detected Failure conditions are met.
	InitCondFail	14	BOOL	Enables to confirm if initial condition is met.
	AutoResetDis	15	BOOL	Disable Auto-reset
CFGW	-		WORD	Phase configuration word.
	Owner	0	BOOL	Read/write access. Enables to configure whether the phase commands will come from the program (0) or the operator (1).
	Semi	1	BOOL	Read/write access. Enables to configure the phase in automatic (0) or semi-automatic (1) mode.
	Manual	2	BOOL	Read/write access. Enables to configure the phase in automatic/semi-automatic (0) or manual (1) mode. **Bit 1 (SEMI)) / Bit 2 (MANUAL)) / Mode** OFF / OFF / Auto ON / OFF / Semi-Auto - / ON / Manual
	Nextstep	3	BOOL	Read/write access. Enables to confirm the change of steps to the next phase step in Semi-Automatic mode. The DFB sets the signal to 0 after it is processed.
	Scroll_Up	5	BOOL	Read/write access. Enables to show the steps previous to the one being shown on the current window in manual mode. The DFB sets the signal to 0 after it is processed.
	Scroll_Down	6	BOOL	Read/write access. Enables to show the steps following the one being shown on the current window in manual mode. The DFB sets the signal to 0 after it is processed.
	Button1	14	BOOL	User configurable control button in phase and Supervision function.
	Button2	15	BOOL	User configurable control button in phase and Supervision function.

Name	Bit Rank Name (if any)	Bit Number (if any)	Type	Description
CStep	-		INT]	Current step.
Reserved	-		INT]	Reserved
CStepD	-		String [22]	Current step description (three characters for step number concatenated with seventeen characters for description).

PhaseGP_CFG_DDT Type

Name	Type	Description
NStep	INT	Number of the new step at which the sub phase needs to be positioned in manual mode. The DFB sets this variable to 0 after it is processed.
StepD	STRING[122]	Description of steps. Limited to six steps. In automatic and semi-automatic mode, it shows the destination steps specified in the current step transitions. In manual mode, it shows the list of the steps defined in the phase. This variable format is: six steps, twenty characters each. Of the twenty characters of each step, the first three are for the step number and the next seventeen are for its description. The value of this variable is assigned for the auxiliary functions.
TransD	STRING[122]	Description of transitions. Limited to six transitions. This variable format is six transitions, twenty characters each. The twenty characters are for its description. The value of this variable is assigned for the auxiliary functions.

Supervision

Overview

This section describes the Supervision resources and runtime services that are available for the management of PhaseGP.

Genies

Genie Properties

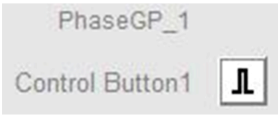
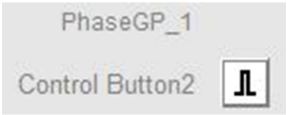
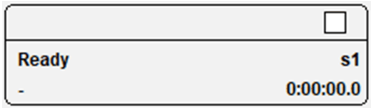
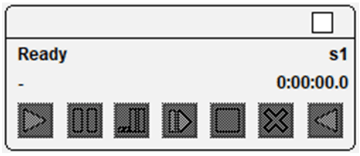
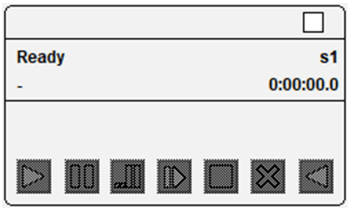
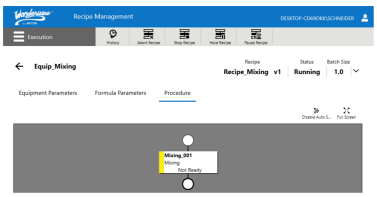
Refer to Genie Properties, page 62.

Genie Icons

Refer to Genie Icons, page 63.

Representation

The table describes the genies of the gpl_phase library:

Graphic symbol	Genie name	Description
	*Controlbutton1*	Genie to supervise a phase status. The label indicates the name of the entity that is being supervised.
	*Controlbutton2*	
	*Panelstate*	
	*Panelstatewithbuttons*	
	*Panelall*	
	*Recipebrowse*	Genie to supervise the recipe management.

Configuring the Recipe Manager Link

Customizing What is Shown in the Web Client

Using a browser control with the typical Recipe Manager Plus URL allows you to show the normal web client within the InTouch HMI. However, you may want to show only the Execution view or even a subset of the Execution view. You would use the flexible execution component to do this.

You embed the flexible execution component in another application window, such as an InTouch application window. The component is still web-based and is driven by customizing the URL. In general, all the functionality of the Execution detail pages is available, but the functionality can be customized by passing various URL query parameters. The action bar is not available for the summary pages. The following are the key points:

- The base URL brings up the typical summary view. The base URL is:

  https://<hostname>/RecipeManagerPlus/index_exec_comp.cshtml

  NOTE: If the SSL Certificate used by Recipe Manager Plus is Issued to Full y QualifiedDomain Name (FQDN), the URL will be https://<myserver.mydomain.com>/RecipeManagerPlus/index_exec_comp.cshtml

- The component operates with the summary and detail view or just the detail view. You can further customize the detail view to restrict functionality and use less screen space.

- It is similar to the web application in functionality but removes the header and menu.
- Optional arguments can be passed to the URL to control features. You can substitute a short name for the full argument name in the URL. For example, the InTouch HMI only supports 131 characters in a URL.

Argument name	Short name	Default	Description
EquipmentName	en	no default	If specified, the detail view appears. The summary view is not available and there is no **Back** button.
AllowEditParameter	aep	true	If set to false (or 0), the **Edit** button is not available in the grid.
ShowActionBar	sab	true	If set to false (or 0), the action bar with the **Assign Formula**, **Download Formula** and **History** buttons is not available.
ShowDetails	sd	true	If set to false (or 0), the Equipment details page is not available. If the user is on summary page and show details (sd) is set to false then the user can navigate to the Equipment details page and can also see the details and the back button will also be available.
ShowEquipmentParameters	sep	true	If Show Equipment Parameters (sep) is set to false (or 0), the Equipment Parameter grid is not available in the details page.
ShowFormulaParameters	sfp	true	If set to false (or 0), the Formula Parameter grid is not available in the details page.
ShowProcedure	sp	true	If Show Procedure (sp) is set to false (or 0), the Procedure tab is not available in the execution details page.
ShowRecipeControls	src	true	If set to false (or 0), all the action buttons for Recipe and Formula are not available.
ShowCapabilityControls	scc	true	If set to false (or 0), all the action buttons for Capability are not available.
AllowViewHistory	avh	true	If set to false (or 0), the History option is not available even if there are Formula downloads and the current user has permission to view reports.
ShowGlobalAlert	sga	true	If set to false (or 0), the global alert icon with the operator action required count is not available on the top right side of the page.

Important:

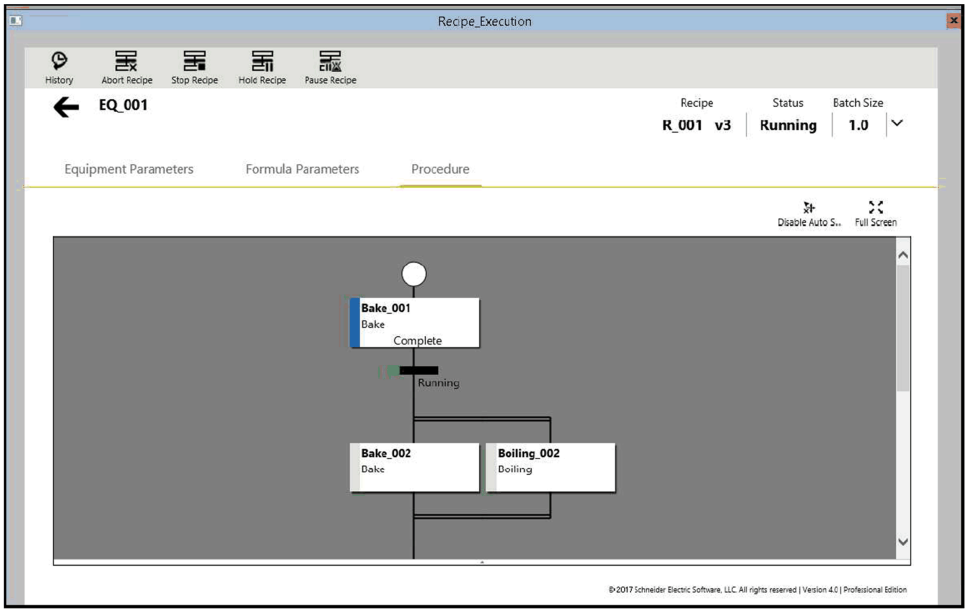
- If only one of the options Show Equipment Parameters, Show Formula Parameters, and Show Procedure is set to true, tab controls will not be available, and only one page is shown.
- If users assign a Formula to the Equipment, the tab controls will not be available because only one page, Equipment Parameters, is shown.
- If users have the necessary privileges, they can perform all operation except assign Formula in the SFC full screen.
- Users cannot adjust the height of the SFC panel and the grid panel when the Recipe is under execution. They can only collapse and expand the grid view panel.
- If users open the user interface by providing the Equipment name in the URL, the global count is the total number of Capabilities waiting for acknowledgment in that Equipment. In the **Operator Actions** slide-in window, only Capabilities waiting for acknowledgment in that Equipment is shown. The **Navigate** to **Execution Summary** button is not available in this case.

User privileges also enable or disable features. For more information, refer to the Working with Security, page 893.

Example 1

This URL shows the details of a specific equipment.

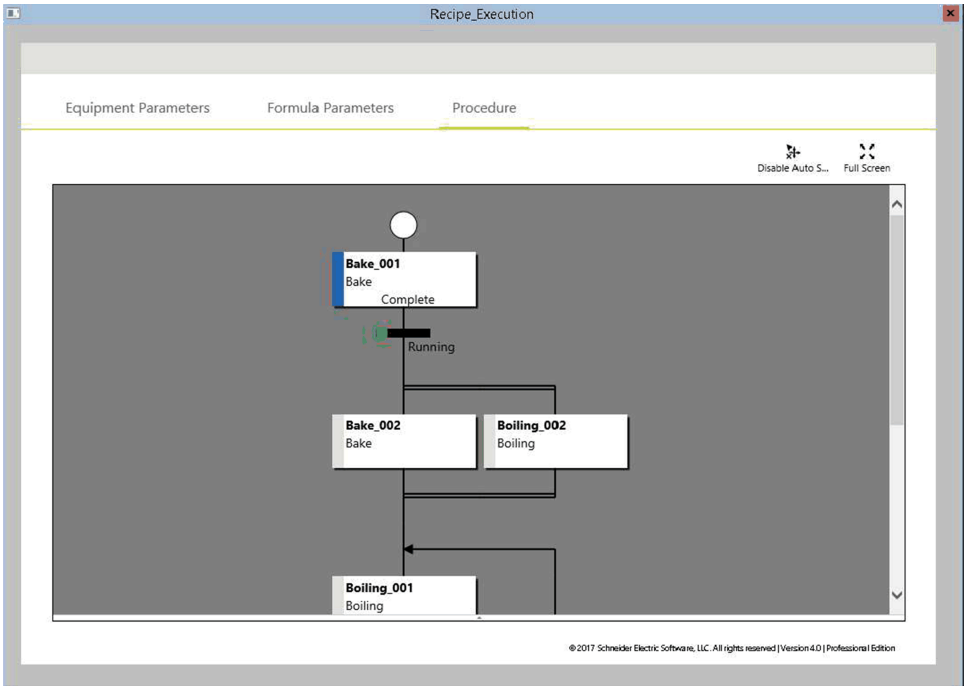
```
#Explorer1.Navigate("https://<hostname>/RecipeManagerPlus/
index_exec_comp.cshtml?en=EQ_001");
```



Example 2

This URL hides the action bar and equipment details.

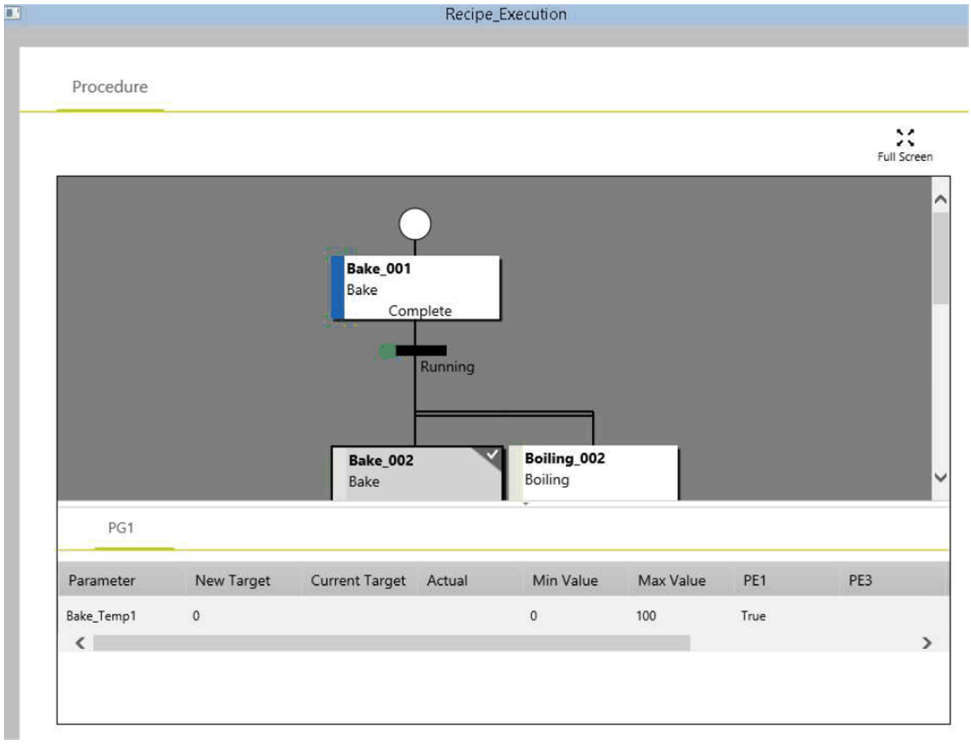
```
#Explorer1.Navigate("https://<hostname>/RecipeManagerPlus/
index_exec_comp.cshtml?en=EQ_001&sab=0&sd=0");
```



Example 3

This URL hides the action bar, equipment details, equipment parameters, formula parameters and global action bar.

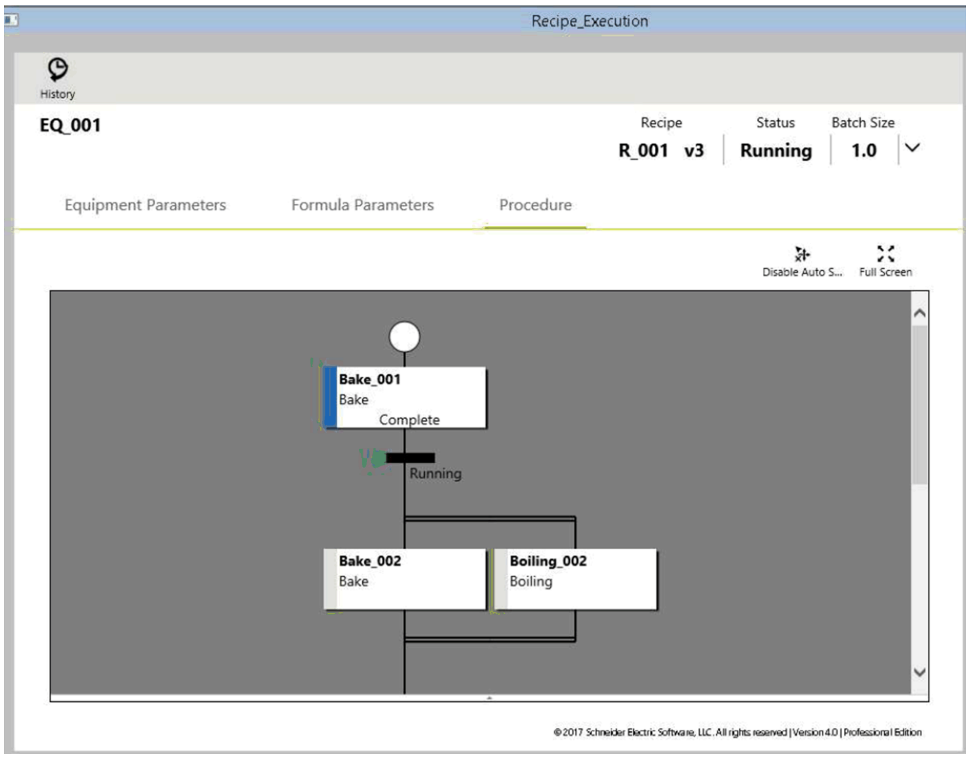
```
#Explorer1.Navigate("https://<hostname>/RecipeManagerPlus/
index_exec_comp.cshtml?en =EQ_001&sab=0&sd=0&sep=0&sfp=
0&sga=0");
```



Example 4

This URL hides the recipe and capability controls.

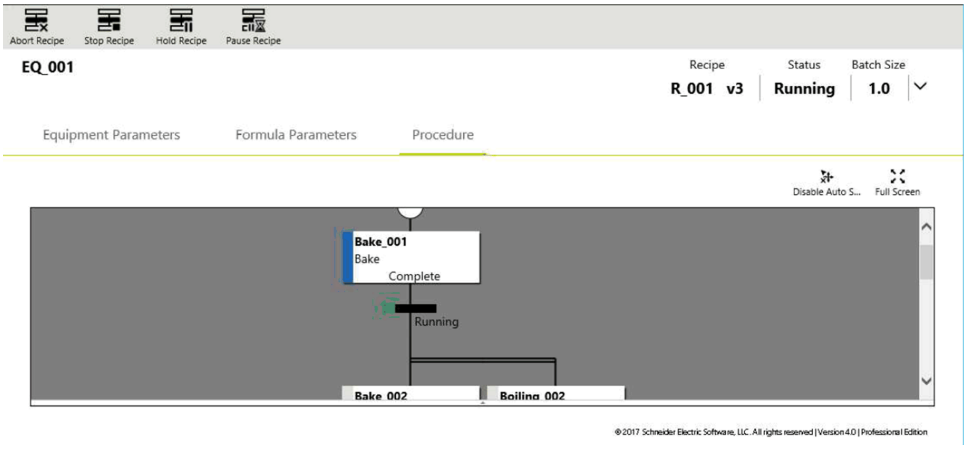
```
#Explorer1.Navigate("https://<hostname>/RecipeManagerPlus/
index_exec_comp.cshtml?en=EQ_001&src=0&scc=0");
```



Example 5

This URL hides the history control.

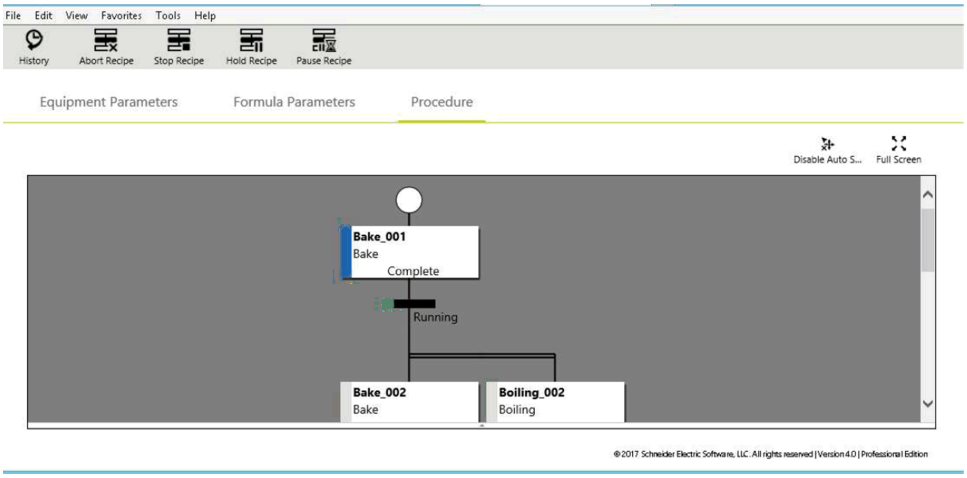
```
#Explorer1.Navigate("https://<hostname>/RecipeManagerPlus/
index_exec_comp.cshtml?en=EQ_001&avh=0");
```



Example 6

This URL hides the Equipment control.

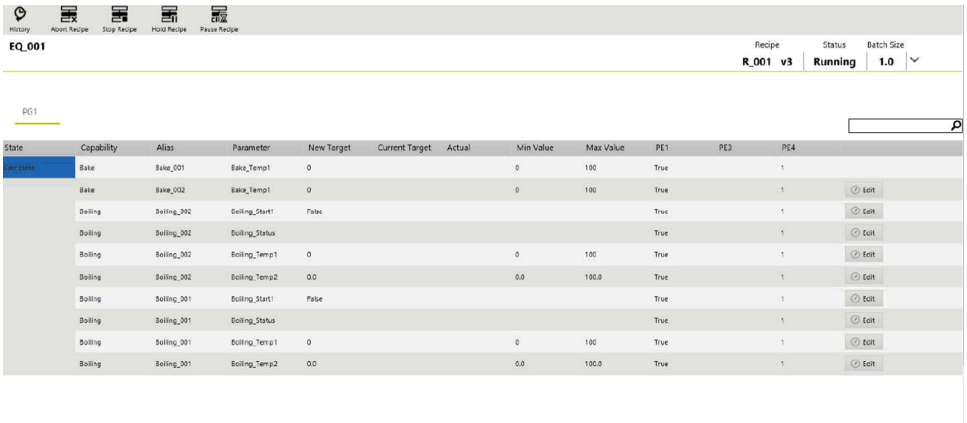
```
#Explorer1.Navigate("https://<hostname>/RecipeManagerPlus/
index_exec_comp.cshtml?en=EQ_001&sd=0");
```



Example 7

This URL hides the **Equipment Parameters**, **Formula Parameters**, and **Procedure** tabs.

```
#Explorer1.Navigate("https://<hostname>/RecipeManagerPlus/
index_exec_comp.cshtml?en=EQ_001&&sep=0&sfp=0&sp=0");
```

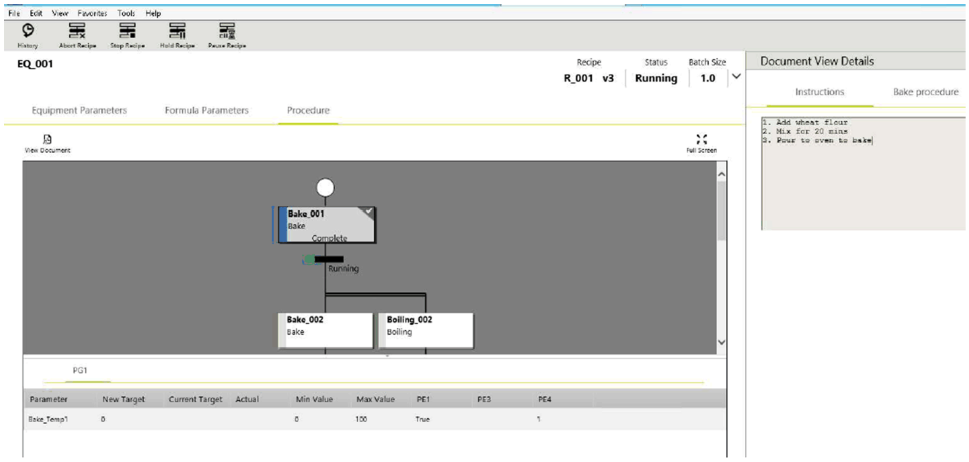


Viewing the Capability Instructions or Documents

You can view the instructions or documents for a Capability which are configured for a Capability instance in a Procedure.

The View Document option is available when you click the Capability instance in the Procedure. Clicking the View Document button displays the instructions in the

sliding window. You can have the View URL option in a new tab depending on the configuration.



Viewing History

If Formulas have been downloaded to Equipment previously or a Recipe is executed and you have permission to view reports, when you access the **Execution** page the **History** icon is visible. You can click the icon to navigate to the Equipment page to view information about the previous downloads or the Recipe execution. Click the back button to return to the **Execution** page.

Faceplate

Representation of Supervision Data

At the beginning of this document, you can find a of the graphic elements that are used in the faceplate.

Available Tabs

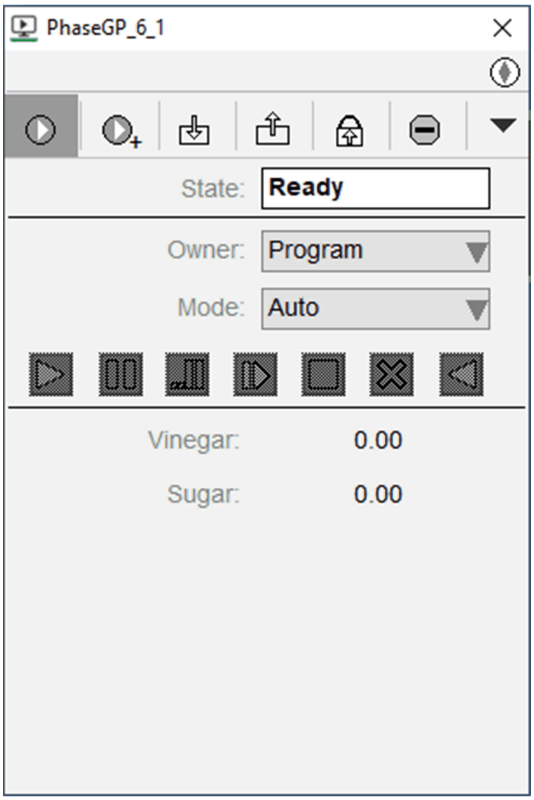
During operation, clicking a phase genie opens a faceplate with the following tabs of PhaseGP:

- Tabs for core functions:
  - Operation
  - Extended Operator, page 914
  - Input Parameters, page 915
  - Output Parameters, page 916
  - State Machine, page 918
- Tabs for optional functions, which appear only if configured:
  - Initial Condition, page 917
  - Failure, page 78

    NOTE: The current detected fail status on the faceplate appears, when user maps detected fail status in 13th bit of `STW` to obtain information in SCADA.

Operator Tab

The figure shows an operator tab:

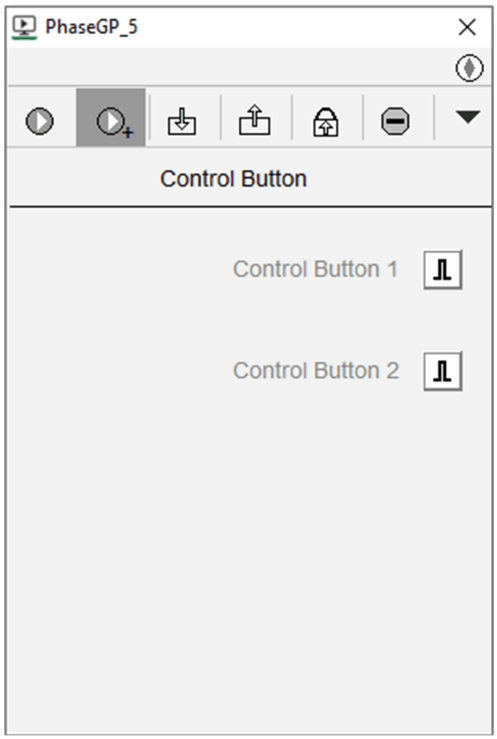


The table describes the control function of an operator tab:

Label	Item name	Description	Security parameter, page 893
State	State	Current state of PhaseGP.	—
Owner	OwnerSelect	Current owner of PhaseGP.	—
Mode	Mode	Allows you to view and change the mode.	—

Extended Operator Tab

The figure shows an Extended Operator tab:

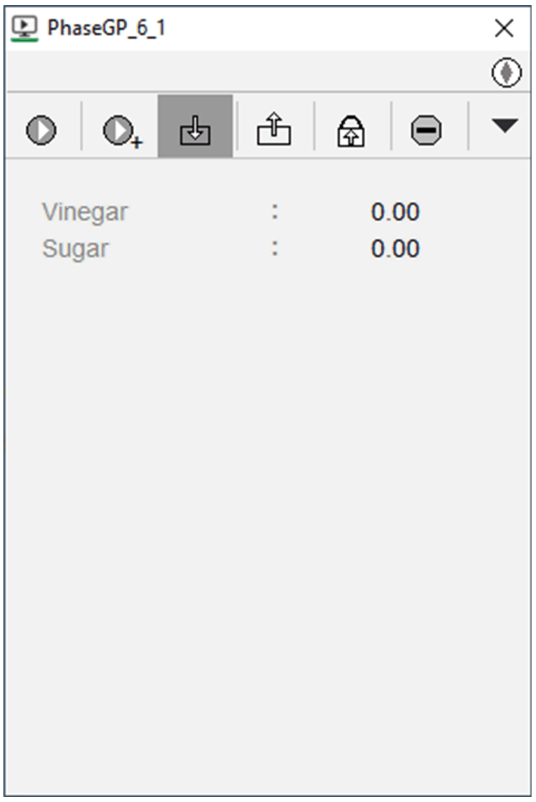


NOTE:

- On clicking the control buttons on Extended Operator tab or Genies, Pulse command is issued to CFGW in corresponding instance of control block.
- Current state (On/Off) will not be displayed.

Input Parameters Tab

The figure shows a Input Parameters tab:

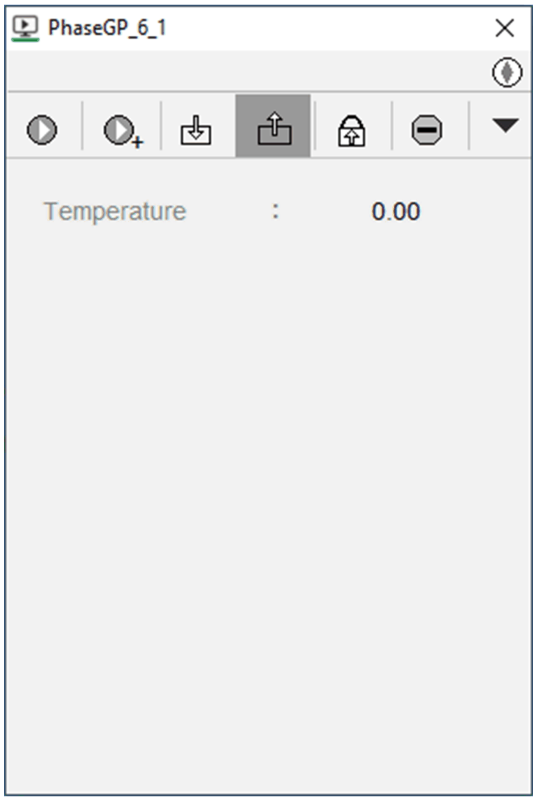


The table describes the lists of items of each groups of Input parameters tab:

Label	Item name	Description	Security parameter, page 893
Input Parameter			
User defined descriptions	IP01 to IP16	Input parameters will get generate according to the `Maximum Phase Parameters` selected in Configuration, page 892.	–

Output Parameters Tab

The figure shows a Output Parameters tab:



The table describes the lists of items of each groups of Output parameters tab:

Label	Item name	Description	Security parameter, page 893
Output Parameter			
User defined descriptions	OP01 to OP16	Output parameters will get generate according to the `Maximum Phase Parameters` selected in Configuration, page 892.	–

Initial Condition Tab

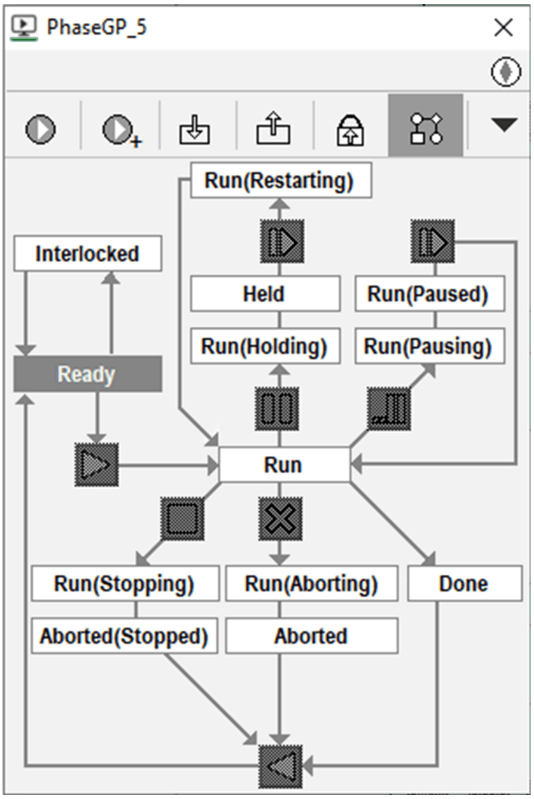
The figure shows a Initial Condition tab:

The Initial Conditions tab displays all conditions which are required to start the phase.

For more details, refer to interlock, page 87.

State Machine Tab

The figure shows a State Machine tab:



The state diagram allows the operator to execute available commands, using the buttons. Label of inactive (passive) states appears in white color and active states appears in grey color.

Items for InBatch Legacy

Overview

This section describes the variables, equipstatus and equipstate variable, abnormal and action required conditions, alarm items, events, and trend item of `Phase`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`Transition {x}`	{y} transition status	Done/Pending	*Bit {z}*
NOTE: • {x} represents the group, range from 1 to 6. • {y} represents the condition, range from 1 to 6. • {z} represents the condition, range from 0 to 5.			
`FailActive`	Fail condition	Active/Inactive	*Bit13*
`InitCondActive`	Initial detected fail condition	–	*Bit14*
`AutoResetDisa-ble`	Disable auto reset	–	*Bit15*

Item name	Description	Enumeration	Address
Bit items derived from StateWord			
Ready	Ready state of Phase	On/Off	*Bit0*
Running	Running state of Phase		*Bit1*
Held	Held state of Phase	–	*Bit2*
HeldForError	Status reached after the hold sequence is completed and the next steady status is reached to a detected failure.	–	*Bit3*
Stopped	The stop sequence is completed. Requires an initialization command to go to Ready status.	–	*Bit4*
Paused	Pause sequence is completed and the next steady status is reached	–	*Bit5*
Aborted	An aborting sequence has been completed. Requires an initialization command -Reset to go to Ready status.	–	*Bit6*
Done	Normal operation has finished. Requires an initialization order -Reset to return to the Idle status.	–	*Bit7*
Holding	A hold command is received and the sequence that allows the process to be changed to a known status is executed. After this sequence is finished, it automatically goes to the held status.	–	*Bit8*
HoldingForError	A detected failure is automatically detected in the process, for example, a detected failure in a Equipment Module and the sequence that allows the process to be to a known status is executed. After this sequence is finished, it automatically goes to	–	*Bit9*
Restarting	Restart command is received while the sequence is being held. The sequence runs to return to the normal operation sequence (Running). The normal operation status is set automatically after the sequence is complete.	–	*Bit10*
Pausing	A Pause command is received for a short period while the normal operation was being executed. The normal sequence runs until it automatically goes to Paused status	–	*Bit11*
Stoping	An order to Stop has been received. The sequence to finish the sequence in a controlled way is executed. The stopped operation status is automatically activated after the sequence is complete.	–	*Bit12*
Aborting	An order to end -Abort has been received. A fast stop,	–	*Bit13*

Item name	Description	Enumeration	Address
	which is not necessarily controlled is executed. The aborted operation status is automatically activated after the sequence is completed.		
Bit items derived from ConfigurationWord.			
OwnerSelect	Owner of control module	Operator/Program	*Bit0*
SemiAutomatic	Semi automatic mode command	–	*Bit1*
Manual	Manual mode command		*Bit2*
NextStep	Next step command		*Bit3*
ScrollUp	Scroll up command		*Bit5*
ScrollDown	Scroll down command		*Bit6*
Button1	Custom command button 1	–	*Bit14*
Button2	Custom command button 2	–	*Bit15*
Bit items derived from CommandWord.			
Start	Allows the normal operation of the sequence-Running to be started.	–	*Bit0*
Hold	Allows the sequence to be stopped and the stopping sequence to be run.	–	*Bit1*
Restart	Allows the sequence for continuing the operation and resume normal operation -Running	–	*Bit2*
Stop	Allows the sequence like Running, Pausing, Paused,Holding, Held, or Restarting to be stopped and the stopping sequence to be run	–	*Bit3*
Reset	Triggers the transition to idle status -in Complete, Aborted, or Stopped status.	–	*Bit4*
Pause	Allows normal operation of the sequence to be paused from the Running status.		*Bit5*
Resume	Allows normal operation of the sequence to be resumed from the Paused status.		*Bit6*
Abort	Allows the sequence to be aborted in any status except Idle, Complete, Aborting, and Aborted and the aborting sequence to be run.		*Bit7*
IO Items			
StatusWord	Status word	–	Phase_ST.STW
Configuration-Word	Configuration word		Phase_ST.CFGW
ParStatusWord	ParStatus word bit array		PhPar05_ST.STW
ParConfiguration-Word	ParConfiguration word bit array		PhPar05_ST.CFGW
NewStep	New sequential step -MANUAL mode		Phase_CFG.NSTEP
StepDescription	Steps descriptions		Phase_CFG.STEPD

Item name	Description	Enumeration	Address
TransitionDe-scription	Transitions descriptions		Phase_CFG.TRANSD
CurrentStepDe-scription	Current step description		Phase_ST.CSTEPD
Elapsedtime	Elapsed time in the current step (dsec)		Phase_ST.ETIME
State	Current phase state		Phase_ST.STATE
Command	Phase commands.		Phase_ST.COMMAND
EquipStateSym-bol	Abnormal state		-
Calculated Items			
EquipState	Current state of the Phase For more details, refer to EquipState Calculated Variable, page 921.	–	GPL_ PhaseEquipState
OwnerSymbol	Shows abnormal owner condition of the Phase		-
Owner	Current owner of the Phase For more details, refer to calculated variable, page 58.		GPL_OwnerBasic
Abnormal	Shows abnormal condition of the Phase For more details, refer to abnormal conditions, page 922.		GPL_ SeqEMAbnormal
ActionRequired	Shows action required condition of the Phase For more details, refer to Action Required Conditions, page 922.		GPL_ SeqEMActionRe-quired
Failure	Refer to Failure Variable Items, page 61.		
Interlock	Refer to Interlock Variable Items, page 60.		

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	State Value	Enumeration
Active	*2*	Running
Passive	*1*	Idle
	128	Done
	32	Paused
	4	Held
	8	Held For Error
	16	Stopped
	64	Aborted
Active to Passive or Transitioning Active	*1024*	Restarting
Passive to Active or Transitioning Passive	*2048*	Pausing
	256	Holding
	512	Holding for error

Equipment State	State Value	Enumeration
	4096	Stopping
	8192	Aborting

Abnormal Conditions

The table describes the abnormal conditions of Phase:

Item name	Description	Address
AbnormalOwner	Abnormal owner status appears if the current owner is not a normal owner which configured at the instance level.	*Bit0*
AbnormalMode	Abnormal mode status appears if the current operating mode is not normal mode.	*Bit1*
AbnormalState	Abnormal state of operator tab.	*Bit3*
AbnormalOperatorTab	Abnormal status of operator tab. This condition appears if the Interlock bypass is enabled.	*Bit8*
AbnormalFailTab	Abnormal status of failure tab.	*Bit10*
AbnormalInterlockTab	Abnormal status of interlock tab.	*Bit11*
AbnormalSequenceTab	Abnormal status of sequence tab.	*Bit13*
AbnormalInitCondTab	Initial condition tab abnormal	*Bit15*

Action Required Conditions

The table describes the action required conditions of Phase:

Item name	Description	Address
ActionOperatorTab	Action required status of operator tab.	*Bit8*
ActionFailTab	Action required status of failure tab.	*Bit10*
ActionInterlockTab	Action required status of interlock tab.	*Bit11*
ActionSequenceTab	Action required status of sequence tab.	*Bit13*
ActionInitCondTab	Initial condition tab action required	*Bit15*

Alarms

The table describes the advanced alarm items that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expression
HeldForEr-ror	Held for error	*(PHASEGP_STATE = 512)* OR *(PHASEGP_STATE = 8)*

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
StateAlarm	All changes in the Phase status and output.
ModeAlarm	All changes in the mode and owner.
ActionAlarm	All actions by the operator.

Items

No trend item is used by Supervision components for Phase.

Items for InBatch and Recipe Management

Overview

This section describes the variables, equipstatus and equipstate variable, abnormal and action required conditions, alarm items, events, and trend item of Phase.

NOTE: For Recipe Management configuration, refer to Recipe Management User Guide.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
Transition {x}	{y} transition status	Done/Pending	*Bit {z}*
NOTE: • {x} represents the group, range from 1 to 6. • {y} represents the condition, range from 1 to 6. • {z} represents the condition, range from 0 to 5.			
FailActive	Detected fail condition	Active/Inactive	*Bit13*
InitCondActive	Initial detected fail condition	–	*Bit14*
AutoResetDisa-ble	Disable auto reset	–	*Bit15*
Bit items derived from StateWord			
Ready	Ready state of Phase	On/Off	*Bit0*
Running	Running state of Phase		*Bit1*
Held	Held state of Phase	–	*Bit2*
HeldForError	Status reached after the hold sequence is completed and the next steady status is reached to a detected failure.	–	*Bit3*
Stopped	The stop sequence is completed. Requires an initialization command to go to Ready status.	–	*Bit4*
Paused	Pause sequence is completed and the next steady status is reached	–	*Bit5*
Aborted	An aborting sequence has been completed. Requires an initialization command	–	*Bit6*

Item name	Description	Enumeration	Address
	-Reset to go to Ready status.		
Done	Normal operation has finished. Requires an initialization order -Reset to return to the Idle status.	–	*Bit7*
Holding	A hold command is received and the sequence that allows the process to be changed to a known status is executed. After this sequence is finished, it automatically goes to the held status.	–	*Bit8*
HoldingForError	A detected failure is automatically detected in the process, for example, a detected failure in a Equipment Module and the sequence that allows the process to be to a known status is executed. After this sequence is finished, it automatically goes to	–	*Bit9*
Restarting	Restart command is received while the sequence is being held. The sequence runs to return to the normal operation sequence (Running). The normal operation status is set automatically after the sequence is complete.	–	*Bit10*
Pausing	A Pause command is received for a short period while the normal operation was being executed. The normal sequence runs until it automatically goes to Paused status	–	*Bit11*
Stoping	An order to Stop has been received. The sequence to finish the sequence in a controlled way is executed. The stopped operation status is automatically activated after the sequence is complete.	–	*Bit12*
Aborting	An order to end -Abort has been received. A fast stop, which is not necessarily controlled is executed. The aborted operation status is automatically activated after the sequence is completed.	–	*Bit13*
Bit items derived from ConfigurationWord.			
OwnerSelect	Owner of control module	Operator/Program	*Bit0*
SemiAutomatic	Semi automatic mode command	–	*Bit1*
Manual	Manual mode command		*Bit2*
NextStep	Next step command		*Bit3*
ScrollUp	Scroll up command		*Bit5*
ScrollDown	Scroll down command		*Bit6*
Button1	Custom command button 1	–	*Bit14*
Button2	Custom command button 2	–	*Bit15*
Bit items derived from CommandWord.			

Item name	Description	Enumeration	Address
Start	Allows the normal operation of the sequence-Running to be started.	–	*Bit0*
Hold	Allows the sequence to be stopped and the stopping sequence to be run.	–	*Bit1*
Restart	Allows the sequence for continuing the operation and resume normal operation -Running	–	*Bit2*
Stop	Allows the sequence like Running, Pausing, Paused,Holding, Held, or Restarting to be stopped and the stopping sequence to be run	–	*Bit3*
Reset	Triggers the transition to idle status -in Complete, Aborted, or Stopped status.	–	*Bit4*
Pause	Allows normal operation of the sequence to be paused from the Running status.		*Bit5*
Resume	Allows normal operation of the sequence to be resumed from the Paused status.		*Bit6*
Abort	Allows the sequence to be aborted in any status except Idle, Complete, Aborting, and Aborted and the aborting sequence to be run.		*Bit7*
IO Items			
StatusWord	Status word	–	Phase_ST.STW
Configuration-Word	Configuration word		Phase_ST.CFGW
ParStatusWord	ParStatus word bit array		PhPar05_ST.STW
ParConfiguration-Word	ParConfiguration word bit array		PhPar05_ST.CFGW
NewStep	New sequential step -MANUAL mode		Phase_CFG.NSTEP
StepDescription	Steps descriptions		Phase_CFG.STEPD
TransitionDe-scription	Transitions descriptions		Phase_CFG.TRANSD
CurrentStepDe-scription	Current step description		Phase_ST.CSTEPD
Elapsedtime	Elapsed time in the current step (dsec)		Phase_ST.ETIME
State	Current phase state		Phase_ST.STATE
Command	Phase commands.		Phase_ST.COMMAND
EquipStateSym-bol	Abnormal state		-
PHCommand	PHCommand		Phase_ST. PhCommand
PHState	PHState		Phase_ST.PhState
CStep	Current Step		Phase_ST.CStep
PHREADY	Phase Ready Staus)		Phase_ST. PhState00
PHRUN	Phase Run Status		Phase_ST. PhState.01

Item name	Description	Enumeration	Address
PHHELD	Phase Held Status		Phase_ST. PhState.02
PHABORTED	Phase Aborted Status		Phase_ST. PhState.06
PHDONE	Phase Done Status		Phase_ST. PhState.14
PHSTART	Phase Start Command		Phase_ST. PhCommand.00
PHHOLD	Phase Hold Command		Phase_ST. PhCommand.01
PHRESTART	Phase Restart Command		Phase_ST. PhCommand.02
PHRESET	Phase Reset Command		Phase_ST. PhCommand.04
PHABORT	Phase Abort Command		Phase_ST. PhCommand.07
PHINTERLOCKED	Phase Interlocked Status		Phase_ST. PhState.15
ControlButton1	Phase Control Button 1		Phase_ST.CFGW.14
ControlButton2	Phase Control Button 2		Phase_ST.CFGW.15
PHHELDFORER-ROR	Phase Held for Error Status		Phase_ST. PhState.03
PHSTOPPED	Phase Stopped Status		Phase_ST. PhState.04
PHPAUSED	Phase paused Status		Phase_ST. PhState.05
PHHOLDING	Phase Holding Status		Phase_ST. PhState.08
PHHOLDINGFOR-ERROR	Phase Holding for error Status		Phase_ST. PhState.09
PHRESTARTING	Phase Restarting		Phase_ST. PhState.10
PHPAUSING	Phase Pausing		Phase_ST. PhState.11
PHSTOPPING	Phase Stopping		Phase_ST. PhState.12
PHABORTING	Phase Aborting		Phase_ST. PhState.13
PHPAUSE	Phase Pause Command		Phase_ST. PhCommand.05
PHRESUME	Phase Resume Command		Phase_ST. PhCommand.06
PHSTOP	Phase Stop Command		Phase_ST. PhCommand.03
PhaseStart	Phase Start Command		Phase_ST. PhaseStart
PhaseHold	Phase Hold Command		Phase_ST. PhaseHold
PhaseReset	Phase Reset Command		Phase_ST. PhaseReset
PhaseResume	Phase Resume Command		Phase_ST. PhaseResume
PhaseRestart	Phase Restart Command		Phase_ST. PhaseRestart
PhaseStop	Phase Stop Command		Phase_ST. PhaseStop
PhasePause	Phase Pause Command		Phase_ST. PhasePause
PhaseAbort	Phase Abort Command		Phase_ST. PhaseAbort

Item name	Description	Enumeration	Address
PhaseReady	Phase Ready Status		Phase_ST. PhaseReady
PhaseRun	Phase Run Status		Phase_ST. PhaseRun
PhaseHeld	Phase Held Status		Phase_ST. PhaseHeld
PhaseHeldForError	Phase Held For Error Status		Phase_ST. PhaseHeldForError
PhaseStopped	Phase Stopped Status		Phase_ST. PhaseStopped
PhasePaused	Phase Paused Status		Phase_ST. PhasePaused
PhaseAborted	Phase Aborted Status		Phase_ST. PhaseAborted
PhaseHolding	Phase Holding Status		Phase_ST. PhaseHolding
PhaseHoldingForEr-ror	Phase Holding For Error Status		Phase_ST. PhaseHoldingFor-Error
PhaseRestarting	Phase Restarting Status		Phase_ST. PhaseRestarting
PhasePausing	Phase Pausing Status		Phase_ST. PhasePausing
PhaseStopping	Phase Stopping Status		Phase_ST. PhaseStopping
PhaseAborting	Phase Aborting Status		Phase_ST. PhaseAborting
PhaseDone	Phase Done Status		Phase_ST. PhaseDone
PhaseInterlocked	Phase Interlocked Status		Phase_ST. PhaseInterlocked
Calculated Items			
EquipState	Current state of the Phase For more details, refer to EquipState Calculated Variable, page 921.	–	GPL_ PhaseEquipState
OwnerSymbol	Shows abnormal owner condition of the Phase		-
Owner	Current owner of the Phase For more details, refer to calculated variable, page 58.		GPL_OwnerBasic
Abnormal	Shows abnormal condition of the Phase For more details, refer to abnormal conditions, page 922.		GPL_ SeqEMAbnormal
ActionRequired	Shows action required condition of the Phase For more details, refer to Action Required Conditions, page 922.		GPL_ SeqEMActionRe-quired
Failure	Refer to Failure Variable Items, page 61.		
Interlock	Refer to Interlock Variable Items, page 60.		

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	State Value	Enumeration
Active	*2*	Running
Passive	*1*	Idle
	128	Done
	32	Paused
	4	Held
	8	Held For Error
	16	Stopped
	64	Aborted
Active to Passive or Transitioning Active	*1024*	Restarting
Passive to Active or Transitioning Passive	*2048*	Pausing
	256	Holding
	512	Holding for error
	4096	Stopping
	8192	Aborting

Abnormal Conditions

The table describes the abnormal conditions of Phase:

Item name	Description	Address
`AbnormalOwner`	Abnormal owner status appears if the current owner is not a normal owner which configured at the instance level.	*Bit0*
`AbnormalMode`	Abnormal mode status appears if the current operating mode is not normal mode.	*Bit1*
`AbnormalState`	Abnormal state of operator tab.	*Bit3*
`AbnormalOperatorTab`	Abnormal status of operator tab. This condition appears if the Interlock bypass is enabled.	*Bit8*
`AbnormalFailTab`	Abnormal status of `failure` tab.	*Bit10*
`AbnormalInterlockTab`	Abnormal status of interlock tab.	*Bit11*
`AbnormalSequenceTab`	Abnormal status of sequence tab.	*Bit13*
`AbnormalInitCondTab`	Initial condition tab abnormal	*Bit15*

Action Required Conditions

The table describes the action required conditions of Phase:

Item name	Description	Address
`ActionOperatorTab`	Action required status of operator tab.	*Bit8*
`ActionFailTab`	Action required status of `failure` tab.	*Bit10*
`ActionInterlockTab`	Action required status of interlock tab.	*Bit11*
`ActionSequenceTab`	Action required status of sequence tab.	*Bit13*
`ActionInitCondTab`	Initial condition tab action required	*Bit15*

Alarms

The table describes the advanced alarm items that are used by Supervision components. It also indicates the address that is configured in the Supervision components to read the corresponding bit of the status word:

Item name	Description	Expression
HeldForEr-ror	Held for error	*(PHASEGP_STATE = 512)* OR *(PHASEGP_STATE = 8)*

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
StateAlarm	All changes in the Phase status and output.
ModeAlarm	All changes in the mode and owner.
ActionAlarm	All actions by the operator.

Items

No trend item is used by Supervision components for Phase.

Auxiliary Functions of PHASEGP

What's in This Chapter

PHTRANSGP 930
PHFAILEDGP 932
PHSTABLESTEPGP 933
PHSTEPDESCGP 934

Overview

This section describes the function blocks that supplement the PHASEGP DFB and that have the objective of facilitating the programming of the control phase.

PHTRANSGP

Overview

This section describes the PHTRANSGP DFB.

Description

General

The main objective of PHTRANSGP DFB is to set up the control phase transitions based on the condition to be evaluated and the next step to be executed when this expression is fulfilled.

Additionally, the DFB manages the information necessary to graphically represent the state of the transitions and steps following the current step (Dest and DestW fields of the PhaseGP_ST_DDT phase state structure).

Include call to this function in the logic of the steps (normally in ST language) to program the transitions between the phase steps.

Function Call

The function call is carried out from the steps of the control subphases (Running, Restarting, Holding, and so on) normally in ST language to program the transitions between these steps.

If several PHTRANSGP calls are included in the same step and if more than one of them is fulfilled in the same step execution cycle, the step corresponding to the transition that is fulfilled and was programmed first will run. The number of PHTRANSGP calls from one single step is limited to 6.

```
TRANS (CONDITION := (*BOOL*),

DESC := (*string[22]*),

NEXTSTEP := (*INT*),

PhaseGP_SC := (*PhaseGP_SC_DDT*)

PhaseGP_ST := (*PhaseGP_ST_DDT*),

PhaseGP_CFG := (*PhaseGP_CFG_DDT*))
```

A function call example is included below:

4: (Wait for end of reactive dosing)

```
if PHSC.ONENTRY then
BISEN2_REACT.REM:=false;
BISEN2_REACT.LSP:=true;
end_if;
BISEN2_REACT.LSPSEL:=(OP01 < (IP01 - 10.0)); (Speed 1 or 2)
TRANS ((IP01 - COLA_REACT) <= OP01, 'Dosing finished?', PHSC.
CSTEP+1, PHSC, PHST, PHCFG);
```

In the previous example, the TRANS call included in step 4 of the phase corresponds to a `PHTRANSGP` block instance to which the following parameters are passed:

- `(IP01 - COLA_REACT) <= OP01OS`: Corresponds to the expression of the condition that needs to be fulfilled for the transition to be executed. You can include complex expressions that result in a Boolean value (true or false).
- `Dosing finished?`: Corresponds to the description of the transition.
- `PHSC.CSTEP+1`: Corresponds to the next step that needs to be executed when the transition is fulfilled. In this case, the next step (step 5) is executed after the current step (step 4). You can include any expression jumping to an existing step in the phase.
- `PHSC`: Corresponds to the `SC` public variable of the `PHASEGP` block instance that is controlling the phase execution.
- `PHST`: Corresponds to the `DDT_ST` variable of the `PHASEGP` block instance that is controlling the phase execution.
- `PHCFG`: Corresponds to the `DDT_CFG` variable of the `PHASEGP` block instance that is controlling the phase execution.

Inputs

Input Parameter Description

Parameter	Type	Description
`Condition`	`BOOL`	1 = Indicates that the condition for executing the transition is fulfilled.
`Desc`	`STRING[22]`	Provides the description of the transition to be represented in the monitoring program. The character string is limited to 20 characters.
`NextStep`	`INT`	Sub phase step number to which a jump needs to be made when the transition is fulfilled. Refer to the `Condition` input. The permissible values for the steps go from 1 to 999 with the exception of `NextStep` to -1. This indicates that when the transition is fulfilled, the corresponding sub phase (to go, for example, from Running to Complete, from Holding to Held, and so on) will be completed.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
PhaseGP_SC	PhaseGP_SC_DDT	Corresponds to the SC public variable of the PHASEGP block instance that is controlling the phase execution.
PhaseGP_ST	PhaseGP_ST_DDT	Provides the data necessary to communicate with the PHASEGP block that is controlling the phase. Refer to the PHASEGP block (see EcoStruxure™ Process Expert , Process Control Services User Guide) for more details.
PhaseGP_CFG	PhaseGP_CFG_DDT	Provides the data necessary to communicate with the PHASEGP block that is controlling the phase. Refer to the PHASEGP block (see EcoStruxure™ Process Expert , Process Control Services User Guide) for more details.

PHFAILEDGP

Overview

This section describes the PHFAILEDGP DFB.

Description

General

The main objective of the PHFAILEDGP DFB is to report detected errors to the phase management block (PHASEGP) during phase execution so that the *HoldingFE* subphase is triggered.

Function Call

The function call is made from the Running, Restarting, Pausing, or Paused control sub phase steps normally in ST language with the purpose of reporting detected errors in the process that is being controlled. The detected error requires the hold phase to be executed (*HoldingFE*).

A function call example is included below:

(Detected failure condition monitoring after initializations)

```
if PHASEGP.CSTEP > 0 and FAILCOND then
FAILED (PHSC); (Trigger execution of the Hold phase)
end_if;
```

In the previous example, the *FAILED* call that is included in the logic monitors the detected error conditions in each execution cycle (in this particular case, but could be associated with a specific step) corresponds to a PHFAILEDGP DFB block instance to which the following parameters are passed:

- PHASEGP: Corresponds to the name of the SC public variable of the PHASEGP block instance that is controlling the execution of the phase.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
PhaseGP_SC	PhaseGP_SC_DDT	Provides the data necessary to communicate with the PHASEGP block that is controlling the phase. Refer to the PHASEGP block (see EcoStruxure™ Process Expert , Process Control Services User Guide) for more details.

PHSTABLESTEPGP

Overview

The section describes the PHSTABLESTEPGP DFB.

Description

General

The main objective of the PHSTABLESTEPGP DFB is to report to the phase management block (PHASEGP) that the current running phase step corresponds to a step, that is, the re-entry point in the normal execution (Running) phase executes after restarting execution (Restarting).

Function Call

The function call is performed from the Running control sub phase steps normally in ST language for the purpose of reporting. The current step is the entry point to which the program should return when the Running sub phase is resumed after finishing the Restarting sub phase. Basically, it enables to resume the phase execution that verifies the appropriate actions are performed or not.

For example, suppose a valve has to be opened. After the valve is opened, the next step involves turning on a pump. While waiting for the valve to open, a detected failure that makes the phase stop occurs (Holding sub phase execution). You can resume execution by running the restart (Restarting) sub phase. After this, the program returns to the normal running sub phase (Running). After the program goes back to this Running sub phase, the intention is to make another attempt at opening the valve (because most likely, it has been closed during the Holding sub phase) not to start the pump in proceeding directly. Therefore, in the valve opening step, the step is marked and does not call the function PHSTABLESTEPGP in the pump starting step.

The following is an example of a function call:

0: (End Water dosing)

```
if PHSC.ONENTRY then

RAMP_AGUA.LSTART:=false;

FIC_AGUA.REM:=false;

FIC_AGUA.MAN:=true;

FIC_AGUA.LOP:=0.0;

VALV_AGUA.LSP:=false;
```

```
STABLESTEP (PHSC);

end_if;

TRANS (VALV_AGUA.LOWPOS, 'Valve Closed?', PHSC.CSTEP+1, PHSC,
PHST, PHCFG);
```

In the previous example, the STABLESTEP call included in the logic that monitors detected failure conditions in each execution cycle (in this particular case, but it could be associated with a specific step) corresponds to a PHSTABLESTEPGP DFB block instance to which the following parameters are passed:

- PHPhase: Corresponds to the name of the SC public variable of the PHASEGP block instance that is controlling the phase execution.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
PhaseGP_SC	PhaseGP_SC_DDT	Provides the data necessary to communicate with the PHASEGP block that is controlling the phase. Refer to the PHASEGP block (see EcoStruxure™ Process Expert , Process Control Services User Guide) for more details.

PHSTEPDESCGP

Overview

This section describes the PHSTEPDESCGP DFB.

Description

General

The main objective of the PHSTEPDESCGP DFB is to assign descriptions to the steps that are defined in the phase to allow you to monitor them from the Supervision system.

Function Calls

The function call is made from the control sub phase steps (Running, Restarting, Holding, and so on) at the beginning of the code to provide the description of the sub phase steps for their representation in the monitoring program. It is normally written in ST language.

Include as many PHSTEPDESCGP calls as there are steps defined in the sub phase or as there are steps that need to be represented in the Supervision system.

The syntax of the function is the following:

```
STEPDESC STEPNUMBER := (INT),

DESC := (string[18]),

PhaseGP_SC := (PhaseGP_SC_DDT),

PhaseGP_ST := (PhaseGP_ST_DDT),
```

```
PhaseGP_CFG := (PhaseGP_CFG_DDT),
```

The following is an example of a function call:

(Running or Pausing Sub phase (302 / 314))

(Step descriptions)

```
STEPDESC (1, 'Init', PHSC, PHST, PHCFG);
STEPDESC (2, 'Valves ON', PHSC, PHST, PHCFG);
STEPDESC (3, 'Wait Valves ON', PHSC, PHST, PHCFG);
STEPDESC (4, 'Dosing Reactive', PHSC, PHST, PHCFG);
STEPDESC (5, 'Motor OFF', PHSC, PHST, PHCFG);
STEPDESC (6, 'Waiting for queue', PHSC, PHST, PHCFG);
```

(Steps and Transitions)

```
case PHSC.CSTEP of
```

1: (#RunningStep0# Initial Step)

(Check initial conditions)

```
if IC_FAIL then
FalloCI := true;
```

`FAILED (PHSC);` (Initial conditions not satisfied)

```
else
FalloCI := false;
```

(Initialization)

In the above example, the STEPDESC calls included at the beginning of the sub phase correspond to a `PHSTEPDESCGP` DFB block instance to which the following parameters are passed in case of the first call:

- 1: Corresponds to step number 1.
- `Init`: Corresponds to the definition of the step.
- `PHSC`: Corresponds to the `SC` public variable of the `PHASEGP` block instance that is controlling the phase execution.
- `PHST`: Corresponds to the `DDT_ST` variable of the `PHASEGP` block instance that is controlling the phase execution.
- `PHCFG`: Corresponds to the `DDT_CFG` variable of the `PHASEGP` block instance that is controlling the phase execution.

Inputs

Input Parameter Description

Parameter	Type	Description
`StepNumber`	`INT`	Number of the sub phase step. The values allowed for the steps are from 1 to 999.
`Desc`	`STRING[22]`	Description of the number of the sub phase step. The character string is limited to 17 characters. This is the value that will be shown in the phase monitoring program.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
PhaseGP_SC	PhaseGP_SC_DDT	Provides the data necessary to communicate with the PHASEGP block that is controlling the phase. Refer to the PHASEGP (see EcoStruxure™ Process Expert , Process Control Services User Guide) block for more details.
PhaseGP_ST	PhaseGP_ST_DDT	Provides the data necessary to communicate with the PHASEGP block that is controlling the phase. Refer to the PHASEGP (see EcoStruxure™ Process Expert , Process Control Services User Guide) block for more details.
PhaseGP_CFG	PhaseGP_CFG_DDT	Provides the data necessary to communicate with the PHASEGP block that is controlling the phase. Refer to the PHASEGP (see EcoStruxure™ Process Expert , Process Control Services User Guide) block for more details.

Public Variables

Public Variable Description

Parameter	Type	Description
OrderNumber	INT	Read-only public variable reserved for building the list of steps in manual mode that will be shown in the monitoring program.
MaxOrderNumber	INT	Read-only public variable reserved for building the list of steps in manual mode that will be shown in the monitoring program.

PHPARxxGP

What's in This Chapter

Description 937
DFB Representation 938
Inputs 938
Outputs 939
Inputs/Outputs 939
Public Variables 940

Overview

This chapter describes the PHPARxxGP DFBs.

Description

General

The PHPAR05, PHPAR10 and PHPAR16 DFBs manage up to 5, 10, and 16 phase parameters (input and output) respectively. The DFBs facilitate the management of parameters from advanced control phases implemented with the General Purpose Library PHASEGP DFB.

These DFB manage the phase parameters with a data structure (DDT) making it easier to implement reusable control phases (the number of pins in a DFB is limited to 32 input pins and 32 output pins) because a single variable is used to provide access to the data that the phase manages.

Function Description

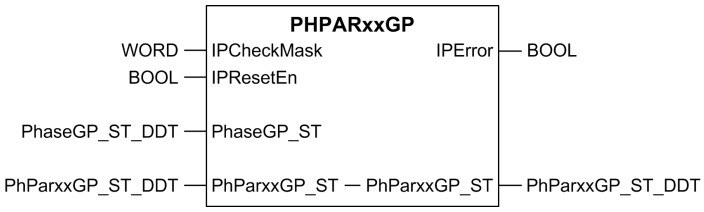
The main functions of the DFB are described in the following table:

Function	Description
Input Parameter Consistency	Optionally, the DFB initializes the values of the input parameters with a value (-0.0) that enables to determine whether or not the applicable parameters of the phase have been correctly downloaded onto the controller. This way, the phase can be stopped from starting with an incomplete or inconsistent set of input parameters.
Owner	The DFB manages control system level, which is the owner (Operator or Program) based on the owner of the associated phase. As a result, it is responsible for defining the input parameters of the phase.
Output Parameters	The DFB manages the output parameters generated from the control phase.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
IpCheckMask	WORD	Optionally, enables to connect a bits word (16) to configure applicable input parameters. Only the applicable parameters will be checked for consistency. Refer to the IPResetEn input and IPError output pin (see EcoStruxure™ Process Expert , Process Control Services User Guide). If no data is connected to this input, the consistency check for input parameter is disabled.
		Bit — **SP** is calculated as:
		0 (2^0) — Input parameter 1 is applicable (1) or not (0).
		1 (2^1) — Input parameter 2 is applicable (1) or not (0).
		... — ...
		15 (2^{15}) — Input parameter 16 is applicable (1) or not (0).
IPResetEn	BOOL	1 = Enables to configure the input parameters need to be initialized with the -0.0 value when the associated control phase switches to an Idle status. This configuration together with the IpCheckMask input pin is used to determine whether or not the input parameters are correctly downloaded (refer to the IPError output pin (see EcoStruxure™ Process Expert , Process Control Services User Guide)).
PhaseGP_ST	PhaseGP_ST_DDT, page 902	Enables to connect the state structure of the control phase for which the parameters need to be managed. The DFB uses this structure to determine the state of the phase (Idle, Running and so on) and its owner (Operator or Program).

Outputs

Output Parameter Description

Parame-ter	Type	Description
IPError	BOOL	1 = A detected error in the assignment. 0 = The check is not activated. The check is only carried out if the IpCheckMask input is connected. If the IpCheckMask input is connected, a check is run to verify whether the applicable parameters as defined in the connected bit mask have a value other than -0.0 or not. The responsibility of assigning the -0.0 value belongs to the DFB or the programmer depending on how the IPResetEn input is configured.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
PhParxxGP_ST	PhParxxGP_ST_DDT	Provides the data that is necessary for managing the types of parameters that the DFB manages input, output, and report.

PhParxxGP_ST_DDT Type

Name	Bit Rank Name (if any)	Bit Number (if any)	Type	Description
STW	–		WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	ErrDwldIPPar	1	BOOL	Detected error downloading input parameter.
CFGW	–		WORD	Provides the means to control the device from the monitoring subsystem. Read/write access to the data contained in this word.
IP01	–		REAL	Read/write access Input parameter 1. Input parameter management: • The DFB initializes the input parameters with a value of -0.0 if the IPResetEn input is enabled when the phase state switches to Idle. • If the owner of the control phase is the Operator (PhaseGP_ST.CFGW.0 = 1), this data is copied to the IPxx fields of the SC public variable. In other words, the input parameters are received from the monitoring or control subsystem in the Operator mode. • If the owner of the control phase is the Program (PhaseGP_ST.CFGW.0 = 0), the input parameters are overwritten with the data in the IPxx fields of the SC public variable.
...	–		REAL	...
IPxx	–		REAL	Read/write access Input parameter xx.
OP01	–		REAL	Read-only access Output parameter 1. Output parameter management:

Name	Bit Rank Name (if any)	Bit Number (if any)	Type	Description
				• The values in the OPxx fields of the SC public variable are continuously copied to these output parameters so that they can be accessed from the monitoring subsystem. Therefore, the control phase of the user needs to define the value of the OPxx parameters of the SC public variable to indicate the results of the execution of the phase.
...	–		REAL	...
OPxx	–		REAL	Read-only access Output parameter xx.

Public Variables

Public Variable Description

Variable	Type	Description
SC	PhParxxGP_SC_DDT	Provides the data needed to manage the parameters of a phase from the control phase itself or from a different external mechanism.

PhParxxGP_SC_DDT Type

Name	BitRank Name (if☐any)	Bit Number (if☐ any)	Type	Description
IP01	–	–	REAL	Read/write access Input parameter 1. Input parameter management: • The DFB initializes the input parameters with a value of -0.0 if the IPResetEn input is enabled when the phase status switches to Idle. • If the owner of the control phase is the Operator (PhaseGP_ST.CFGW.0 = 1), this data is copied to the IPxx fields of the SC public variable. In other words, the input parameters are received from the monitoring or phase control subsystem in the Operator mode. • If the owner of the control phase is the Program (PhaseGP_ST.CFGW.0 = 0), the input parameters are overwritten with the data in the IPxx fields of the SC public variable.
...	–	–		...
IPxx	–	–	REAL	Read/write access Input parameter xx.
OP01	–	–	REAL	Read/write access Output parameter 1. Output parameter management: • The values in the OPxx fields of the SC public variable are continuously copied to these output parameters so that you can access the parameters from the monitoring subsystem. Therefore, the control phase of the user needs to define the value of the OPxx parameters of the SC public variable to indicate the results of the phase execution.
...	–	–		...
OPxx	–	–	REAL	Read/write access Output parameter xx.

Name	BitRank Name (if any)	Bit Number (if any)	Type	Description
Properties	–	–	WORD	Group of boolean datatypes.
	IPError	1	BOOL	Read-only access Refer to the IPError output pin (see EcoStruxure™ Process Expert , Process Control Services User Guide).

UserPhasexxGP

What's in This Chapter

DFB Representation 942
Inputs 942
Inputs/Outputs 943

Overview

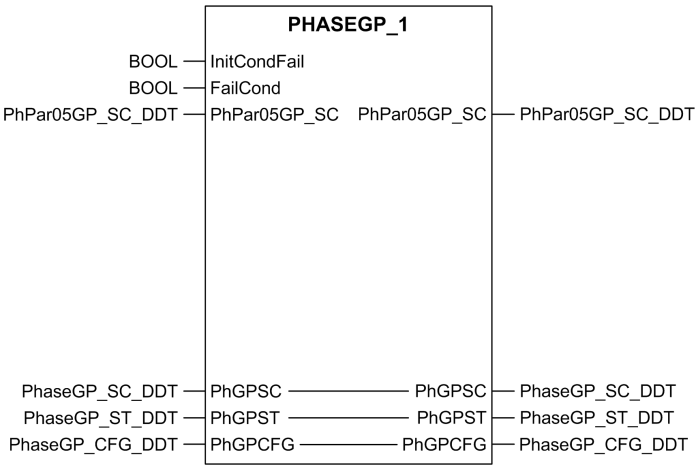
This chapter describes the `UserPhasexxGP` DFBs.

DFB Representation

Representation

The DFB that is used in the program has the following aspect at the section level when imported.You can use it in any of the programming languages, although it is designed for use with the FBD language.

The figure shows an example of the `UserPhasexxGP` DFB instance:



Implement the phase logic that controls the component in ST language to take advantage of the phase control potentials making the phase reusable. That is, you can run simultaneously several instances of the phase that you have implemented. This reduces engineering and maintenance efforts when it is necessary to set up the same control phases in several process units.

Inputs

Input Parameter Description

Parameter	Type	Description
`InitCondFail`	`BOOL`	User can leverage on this input pin to connect to interlock conditions summary (*Condsum*) or other DFBs to take necessary actions depending on the value of `InitCondFail`.
`FailCond`	`BOOL`	User can leverage on this input pin to connect to detected failure conditions summary (*Condsum*) to take necessary actions depending on the value of `FailCond`.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
PhPar05GP_SC	PhPar05GP_SC_DDT, page 940	Provides the data needed to manage the parameters of a phase from the control phase itself or from a different external mechanism.
PhGPSC		Provides the necessary data for background monitoring.
PhGPST	PhaseGP_ST_DDT, page 902	Provides the necessary data for monitoring and controlling phase execution.
PhGPCFG	PhaseGP_CFG_DDT, page 906	Provides the necessary data for background monitoring.

PHASEGP - Example of Use

What's in This Chapter

Description 944
DFB Representation 945
Reusable Phase Logic 945

Overview

This chapter describes the example use of the PHASEGP DFB.

Description

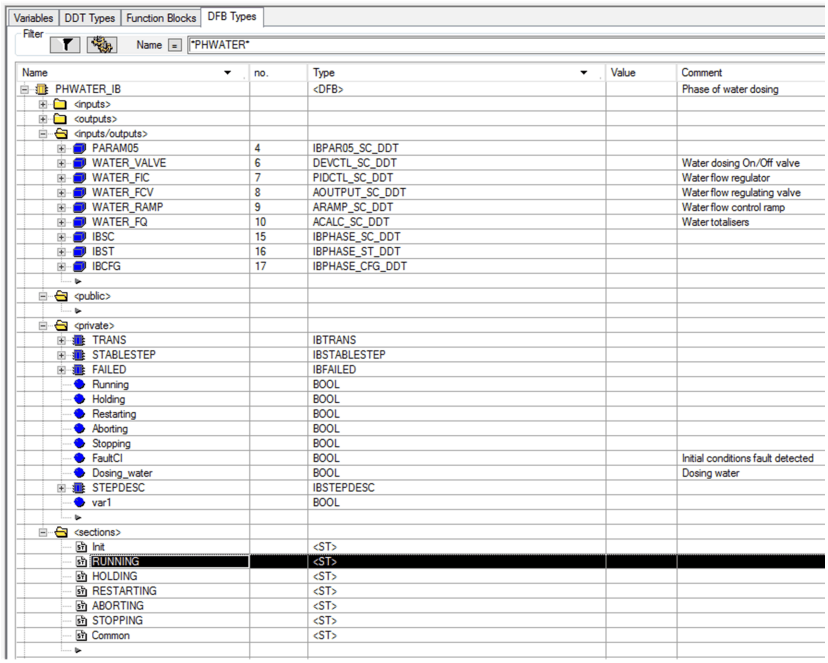
General

This chapter gives an example of phase manager controlled by the PHASEGP, PHTRANSGP, PHSTABLESTEPGP, PHFAILEDGP, and PHSTEPDESCGP DFBs.

The objective of the phase is to control the loading of water in a process unit. For this purpose, a reusable component (DFB) is created so that user can use this phase in different process units by only changing the connections of the component pins.

Reusable Phase Definition

The reusable DFB of the user, PHWATER, declares the PHASEGP_Instance variable to manage the state of the phase.



Variables | DDT Types | Function Blocks | DFB Types

Filter: Name = *PHWATER*

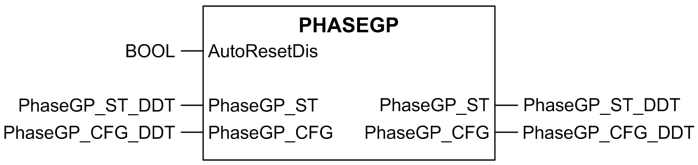
Name	no.	Type	Value	Comment
PHWATER_IB		<DFB>		Phase of water dosing
<inputs>				
<outputs>				
<inputs/outputs>				
PARAM05	4	IBPAR05_SC_DDT		
WATER_VALVE	6	DEVCTL_SC_DDT		Water dosing On/Off valve
WATER_FIC	7	PIDCTL_SC_DDT		Water flow regulator
WATER_FCV	8	AOUTPUT_SC_DDT		Water flow regulating valve
WATER_RAMP	9	ARAMP_SC_DDT		Water flow control ramp
WATER_FQ	10	ACALC_SC_DDT		Water totalisers
IBSC	15	IBPHASE_SC_DDT		
IBST	16	IBPHASE_ST_DDT		
IBCFG	17	IBPHASE_CFG_DDT		
<public>				
<private>				
TRANS		IBTRANS		
STABLESTEP		IBSTABLESTEP		
FAILED		IBFAILED		
Running		BOOL		
Holding		BOOL		
Restarting		BOOL		
Aborting		BOOL		
Stopping		BOOL		
FaultCl		BOOL		Initial conditions fault detected
Dosing_water		BOOL		Dosing water
STEPDESC		IBSTEPDESC		
var1		BOOL		
<sections>				
Init		<ST>		
RUNNING		<ST>		
HOLDING		<ST>		
RESTARTING		<ST>		
ABORTING		<ST>		
STOPPING		<ST>		
Common		<ST>		

NOTE: Declaration of the auxiliary components manages the transitions (PHTRANSGP), steady-step flags (PHSTABLESTEPGP), and notifications to trigger the holding sub phase (PHFAILEDGP).

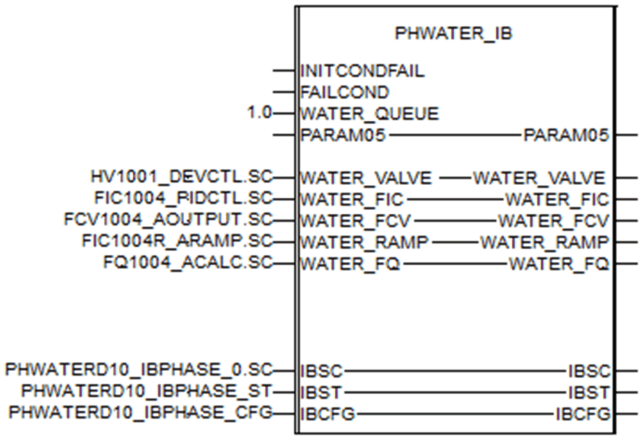
DFB Representation

Representation

The following diagrams include the calls to the phase created by the user as well as to the control block that will control it (PHASEGP).



NOTE: The call to the block controls the execution of the advanced equipment module.



NOTE: The call to the functional block implements the control phase of the user and will reuse for different process units if needed.

Reusable Phase Logic

Phase Logic With Description

The basic structure of the phase as well as the steps that are defined for the phase objective are included: water dosing. The parts of the code that make up the basic structure of the phase are marked in bold.

(Initialization)

The following common actions have to be executed before the phase execution:

```
Running := PHSC.RUNNINGX;

Holding := PHSC.HOLDINGX;

Restarting := PHSC.Restarting;

Aborting := PHSC.Aborting;

Stopping := PHSC.Stopping;

PHSC.INITCONDFAIL := INITCONDFAIL;
```

NOTE: The default actions to be taken by DFB are defined in this section. However, if any section has to be added by user and actions need to be taken before execution of that section, it has to be added here.

(Running or Pausing Subphase)

```
STEPDESC (1, 'Init', PHSC, PHST, PHCFG);
```

```
STEPDESC (2, 'Valves ON', PHSC, PHST, PHCFG);
STEPDESC (3, 'Wait Valves ON', PHSC, PHST, PHCFG);
STEPDESC (4, 'Dosing Water', PHSC, PHST, PHCFG);
STEPDESC (5, 'Valves OFF', PHSC, PHST, PHCFG);
STEPDESC (6, 'Waiting for queue', PHSC, PHST, PHCFG);
```

NOTE: Definition of the descriptions for the subphase steps.

(Steps and Transitions)

```
case PHSC.CSTEP of
```

1: (#RunningStep0# Initial Step)

NOTE: First step with a value of 1.

(Check initial conditions)

```
if INITCONDFAIL then
FaultCI := true;
FAILED (PHSC);
```
(Initial conditions not satisfied)

NOTE: Report detected failure to trigger the holding subphase.

```
else
FaultCI := false;
```

(Initializations)

```
PARAM05.OP01:=0.0;
```

(Reset and water totalizer activation)

```
WATER_FQ.TOTALRST:=true;
WATER_FQ.TOTALEN:=true;
TRANS (TRUE, 'True', PHSC.CSTEP + 1, PHSC, PHST, PHCFG);
```
(Next Step)

NOTE: Unconditional transition to the next step.

```
end_if;
```

2: (Start Water dosing)

```
if PHSC.ONENTRY then
STABLESTEP (PHSC);
```
(In this step, it will initialize the execution in case of Stop/Continue)

NOTE: Execute the actions only for the first time.

NOTE: Flag this step as steady.

```
Dosing_water:=true;
WATER_VALVE.REM:=false;
WATER_VALVE.LSP := true;
if PARAM05.IP01 = 2.0 then
WATER_FCV.REM:=true;
WATER_FIC.REM:=false;
WATER_FIC.MAN:=true;
WATER_FIC.LOP:=10.0;
else
WATER_FCV.REM:=false;
WATER_FCV.LSP:=100.0;
```

```
end_if;
end_if;
TRANS (TRUE, 'True', PHSC.CSTEP + 1, PHSC, PHST, PHCFG);
```
(Next step)

NOTE: The evaluation priority for these transitions is the order in which they are shown (up down; these transitions are not functional, and have only been included as an example in this case). You can define a maximum of 6 transitions.

3: (Wait for open valve)

```
TRANS (WATER_VALVE.HIGHPOS, 'Valve open?', PHSC.CSTEP + 1, PHSC,
PHST, PHCFG);
```
(Next step)

4: (Wait for end of Water dosing)

```
if PHSC.ONENTRY then
if PARAM05.IP01 = 2.0 then
```
(Dose)

```
WATER_RAMP.REM := false;
```
(Set point of flow ramp)

```
WATER_RAMP.LTARGETSP:=PARAM05.IP03;
```
(Desired set point)

```
WATER_RAMP.LSTART:=true;
WATER_FIC.REM:=true;
WATER_FIC.AUTO:=true;
WATER_FIC.LSP:=PARAM05.IP03;
```
(Flow regulation)

```
end_if;
end_if;
TRANS ((PARAM05.IP02 - WATER_QUEUE) <= PARAM05.OP01, 'Dosing
finished?', PHSC.CSTEP + 1, PHSC, PHST, PHCFG);
```

5: (End Water dosing)

```
if PHSC.ONENTRY then
if PARAM05.IP01 = 2.0 then
WATER_RAMP.LSTART:=false;
WATER_FIC.REM:=false;
WATER_FIC.MAN:=true;
WATER_FIC.LOP:=0.0;
else
WATER_FCV.LSP:=0.0;
end_if;
WATER_VALVE.LSP:=false;
STABLESTEP (PHSC);
```
(In this step, it will initialize the execution in case of Stop/Continue)

```
end_if;
TRANS (WATER_VALVE.LOWPOS, 'Valve Closed?', PHSC.CSTEP + 1,
PHSC, PHST, PHCFG);
```

6: (Wait for Water queue)

```
if PHSC.ONENTRY then
```

(Wait for `COLA_AGUA` (Water queue) to fall)

STABLESTEP (PHSC); (In this step, it will initialize the execution in case of Stop/Continue)

end_if;

TRANS (PHSC.ETIME >= 50, 'Waiting for queue', -1, PHSC, PHST, PHCFG); (End of phase)

NOTE: Complete phase (go to COMPLETE) when the transition condition is met.

end_case;

(Actions common to several steps)

(Detected failure condition monitoring after initializations)

if PHSC.CSTEP > 0 and FAILCOND then

FAILED (PHSC); (* It causes Hold phase execution *)

end_if;

(*Holding* Subphase)

STEPDESC (1, 'Init', PHSC, PHST, PHCFG);

STEPDESC (2, 'Closing Valves', PHSC, PHST, PHCFG);

(Steps and Transitions)

case PHSC.CSTEP of

1: (Initial Step)

if FaultCI then

Dosing_water:=false;

end_if;

TRANS (FaultCI, 'Not InitCond', -1, PHSC, PHST, PHCFG); (End of phase)

TRANS (NOT FaultCI, 'InitCond OK', PHSC.CSTEP + 1, PHSC, PHST, PHCFG); (Next step)

2:

if PHSC.ONENTRY then

WATER_VALVE.LSP:= false;

WATER_RAMP.LSTART:=false;

WATER_FIC.MAN:=true;

WATER_FIC.LOP:=0.0; (To close regulating valve)

end_if;

If (Dosing_water) AND PHSC.ETIME >= 50 then

Dosing_water:=false;

end_if;

TRANS (PHSC.ETIME >= 50 OR NOT Dosing_water, 'Waiting for queue', -1, PHSC, PHST, PHCFG); (End of phase)

end_case;

(Restarting subphase)

STEPDESC (1, 'Init', PHSC, PHST, PHCFG);

(Steps and Transitions)

```
case PHSC.CSTEP of
```

1: (Initial Step)

```
TRANS (TRUE, 'True', -1, PHSC, PHST, PHCFG);
```
(End of phase)

```
end_case;
```

(Aborting subphase)

```
STEPDESC (1, 'Closing Valves', PHSC, PHST, PHCFG);
```

(Steps and Transitions)

```
case PHSC.CSTEP of
```

1: (Initial Step)

```
WATER_VALVE.REM:=false;
WATER_VALVE.LSP:=false;
WATER_RAMP.LSTART:=false;
WATER_FIC.MAN:=true;
WATER_FIC.LOP:=0.0;
```
(Close control valve)

```
Dosing_water:=false;
TRANS (TRUE, 'True', -1, PHSC, PHST, PHCFG);
```
(End of phase)

```
end_case;
```

(Stopping subphase)

```
STEPDESC (1, 'Closing Valves', PHSC, PHST, PHCFG);
STEPDESC (2, 'Waiting for queue', PHSC, PHST, PHCFG);
```

(Steps and Transitions)

```
case PHSC.CSTEP of
```

1: (Initial Step)

```
WATER_VALVE.REM:=false;
WATER_VALVE.LSP:= false;
WATER_RAMP.LSTART:=false;
WATER_FIC.MAN:=true;
WATER_FIC.LOP:=0.0;
```
(To close regulating valve)

```
TRANS (TRUE, 'True', PHSC.CSTEP + 1, PHSC, PHST, PHCFG);
```
(Next step)

2:

```
If (Dosing_water ) AND PHSC.ETIME >= 50 then
Dosing_water:=false;
end_if;
TRANS (PHSC.ETIME >= 50 OR NOT Dosing_water, 'Waiting for queue',
-1, PHSC, PHST, PHCFG);
```
(End of phase)

```
end_case;
```

Common actions to be executed after the phase execution.

```
if Dosing_water then
PARAM05.OP01 := WATER_FQ.TOTAL;
end_if;
```

NOTE: No default actions are defined in this section. However, if any action has to be taken after execution of section, it has to be added here.

Advanced Process Control

What's in This Part

Advanced PID 952
Two Loop PID Control 968
Four Loop PID Control 975

Overview

This part provides the functionality of the templates, a detailed description of the control (functions, pins, pin layout, and variables of the function blocks), and the Supervision functions for the advanced process control.

These components do not reflect any specific installation.

⚠ WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate them.
- Review the implications of communication link interruptions and take actions to mitigate them.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, and applicable codes and regulations.
- Apply local accident prevention and safety regulations and guidelines.[1]
- Test each implementation of a system for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

Advanced PID

What's in This Chapter

Template 952
Control – PIDADVCTLGP 954
Control – PIDMUXGP 958
Control – PIDTUNEGP 958
Control – ADVCTLDATA 958
Supervision 961

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the advanced PID with monitoring interface.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$PIDADVGP` template.

This process template helps monitor and control the controller easily from the Supervision system and provides the operating modes used in the remaining EcoStruxure Process Expert process templates

Description

General Description

The `$PIDADVGP` process control module template is used to condition the signals associated with the control of a PID Feed Forward (PIDFF)-type controller.

Functional Description

All the functionalities of `$PIDGP` template are valid for `$PIDADVGP`. For the functional description of the `$PIDADVGP` template, refer to Functional Description, page 519.

The additional advanced functionalities of `$PIDADVGP` template when connected to upstream objects `($AnalogInputGP)` and downstream objects `($AnalogOutputGP, $MotorVSGP, or $ControlValveGP)` are described in the following table:

Function	Description
Downstream object in Program-Local	Advanced PID Output tracks the output of downstream objects.
Downstream object in Operator	
Downstream object in Interlocked	
Downstream object in Program-Remote	Downstream object is controlled by Advanced PID Output (connected to downstream object RSP).
Upstream object in bad state	Advanced PID is forced to manual mode.

Function	Description
Setpoint increment-decrement rate	Setpoint ramping feature is available in Advanced PID. Setpoint Increment-Decrement rate can be configured from supervision.
Output rate	Output rate limit feature is available in Advanced PID. Output rate can be configured from supervision.

Parameters

Control

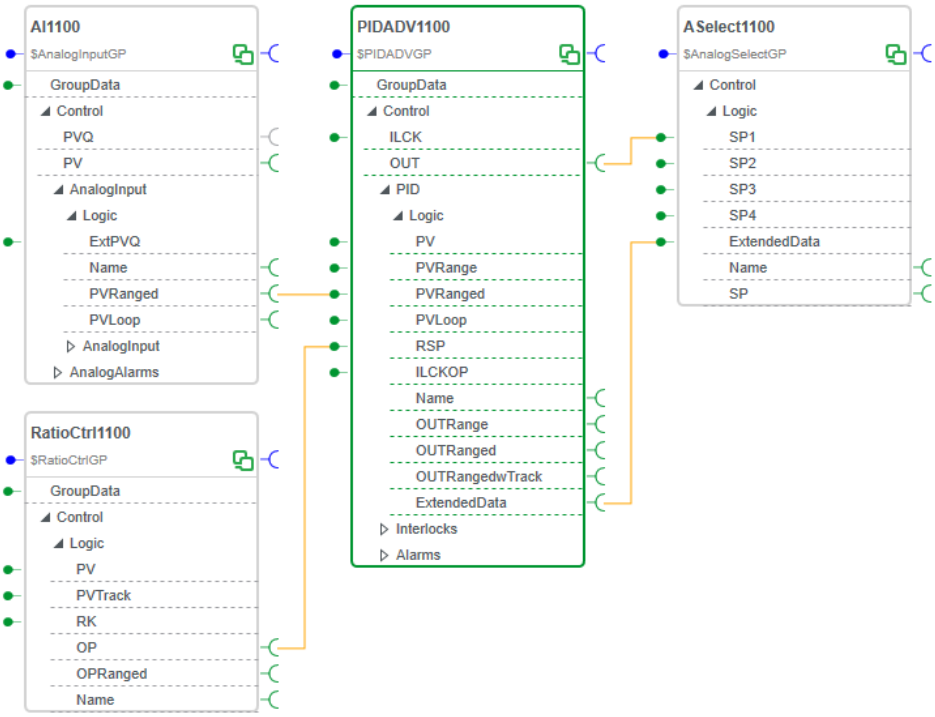
Refer to, page 520, for the Control parameters, .

Supervision

Refer to, page 520, for Supervision parameters.

Interfaces

This figure shows the `$PIDADVGP` template as it appears in `Links Editor` and an example of other templates connected to it:



NOTE: If Multiple Advanced Controllers are connected to Setpoint inputs `(SP1/SP2/SP3/SP4)` of Analog Select block, then all the advanced controllers (except the controller selected from Analog Select) always tracks the downstream object `($AnalogOutputGP/$ControlValveGP/$MotorVSGP)`.

NOTE: For analog input and PIDADVGP faceplate navigation, it is recommended to define the parameter `Process Page Name` where these instances will be located in graphic page else during navigation the process display area will not show the content. For parameter description, refer to, page 43.

The following table lists the link interface that creates link symbol on faceplate:

Interface name	Runtime parameter
PV	PresentValueNavigate
PVRange	
PVRanged	
PVLoop	
RSP	SetpointNavigate
OUTRange	OutputNavigate
OUTRanged	
OUTRangedwTrack	

For more details, refer to Supervision chapter > Faceplate > Navigation Link of General Purpose Library User Guide.

Refer to common control module template interface, page 47 for more details on common interfaces.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control – PIDADVCTLGP

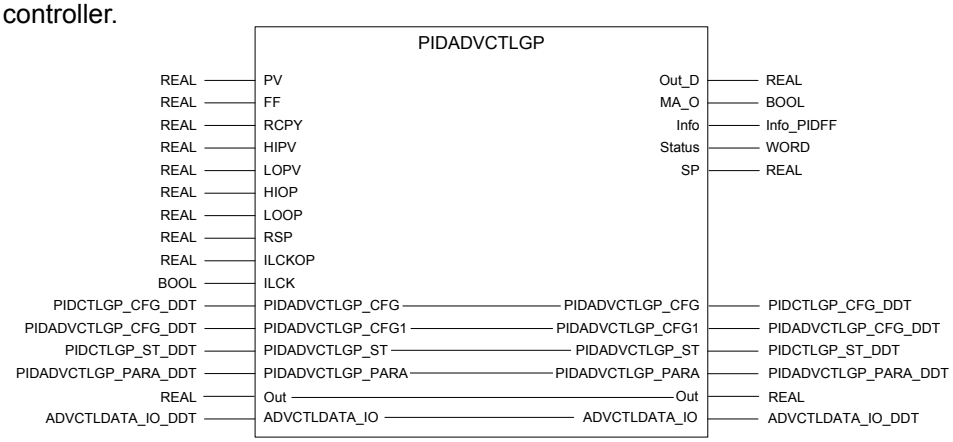
Overview

This section describes the `PIDADVCTLGP` DFB.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Refer to Inputs, page 524.

Outputs

Output Parameter Description

Refer to Outputs, page 525.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
PIDADVCTLGP_CFG	PIDCTLGP_CFG_DDT, page 526	Provides the data necessary to configure the DFB from the monitoring subsystem or the sequential control.
PIDADVCTLGP_CFG1	PIDADVCTLGP_CFG_DDT	Provides the data necessary to configure the DFB from the monitoring subsystem or the sequential control.
PIDADVCTLGP_ST	PIDCTLGP_ST_DDT, page 527	Provides the data necessary to monitor and/or control the DFB status.
PIDADVCTLGP_PARA	PIDADVCTLGP_PARA_DDT	Data structure of the PIDFF controller configuration within the PIDADVCTLGP DFB.
PIDADVTuneGP_ATUNE	PIDTUNEGP_PARA_DDT, page 533	Data structure of the AUTOTUNEGP DFB configuration within the PIDADVCTLGP DFB.
Out	REAL	Output signal of the controller.
ADVCTLDATA_IO_DDT	ADVCTLDATA_IO_DDT	Advance control data structure.

PIDADVCTLGP_CFG_DDT Type

Name	Type	Description
SPIncRate	REAL	Rising gradient in units per second > = 0.
SPDecRate	REAL	Falling gradient in units per second > = 0.
OutRate	REAL	Limit for output in units per second > = 0.
AutoSP	REAL	Current setpoint.

PIDADVCTLGP_PARA_DDT Type

Name	Bit Name	Type	Description
PARA_PIDFF	–	Para_PIDFF	PIDFF parameters allows to populate additional data for PIDFF.
	id	UINT	Reserved for autotuning.
	pv_inf	REAL	Lower limit of the process value range.
	pv_sup	REAL	Upper limit of the process value range.
	out_inf	REAL	Lower limit of the output value range.
	out_sup	REAL	Upper limit of the output value range.
	rev_dir	BOOL	Parameter block for PIDFF: • 0 = opposite action of the PID controller. • 1 = direct action of the PID controller.
	mix_par	BOOL	• 1 = PID controller with parallel structure. • 1 = PID controller with mixed structure.

Name	Bit Name	Type	Description
	aw_type	BOOL	1 =Anti windup halt is filtered.
	en_rcpy	BOOL	1= The RCPY input is used.
	kp	REAL	Proportional action coefficient (gain).
	ti	TIME	Integral time
	td	TIME	Derivative time
	kd	REAL	Differential gain
	pv_dev	BOOL	Type of differential contribution: • 1 = Differential contribution in relation to system deviation. • 0 = Differential contribution in relation to regulating variable (process value).
	bump	BOOL	• 1 = Transition to automatic mode with bump. • 0 = Bumpless transition to automatic mode.
	dband	REAL	Dead zone on deviation.
	gain_kp	REAL	Reducing the proportional contribution within the dead zone dband.
	ovs_att	REAL	Reducing the overrun.
	outbias	REAL	Manual compensation for the static deviation.
	out_min	REAL	Lower limit of the output.
	out_max	REAL	Upper limit of the output.
	outrate	REAL	Limit for output modification in units per second (= 0).
	ff_inf	REAL	Lower limit of the FF range.
	ff_sup	REAL	Upper limit of the FF range.
	otff_inf	REAL	Lower limit of the out_FF range.
	otff_sup	REAL	Upper limit of the out_FF range.
SAMPLETM	–	TIME	Sample time for connected control SAMPLETM controller configuration within the PIDCTL DFB. Refer to the standard SAMPLETM DFB in the *Control* documentation for more details.

ADVCTLDATA_IO_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	USBadSt	*Bit0*	BOOL	Upstream object channel detected failure.
	DSBadSt	*Bit1*	BOOL	Downstream object channel detected failure.
	OPTrack-En	*Bit2*	BOOL	1 = output tracking enable.
	DSAlarm	*Bit3*	BOOL	1 = downstream object alarm.
	DSFailed	*Bit4*	BOOL	1 = downstream object detected failed.
	DSILCKED	*Bit5*	BOOL	1 = downstream object interlocked.
	DSOSS	*Bit6*	BOOL	1 = downstream object out of service.
	DSOwner	*Bit7*	BOOL	1 = downstream owner is operator
TrackValue	–	–	REAL	Track value
USHighPV	–	–	REAL	Upstream PV high range
USLowPV	–	–	REAL	Upstream PV low range

Public Variables

Public Variable Description

Variable	Type	Description
SC	PIDADVCTLGP_SC_DDT	Provides the frequently needed data to monitor and control the regulator status from the sequential control.

PIDADVCTLGP_SC_DDT Type

Name	Type	Description
LSP	REAL	Read/write access Enables to assign the local set-point for the sequential control if the owner is the Program (Owner is 0), the selected mode is AUTO, and the selected set-point is Local (SC.Rem is 0). Otherwise, the current set-point (SP output) is continuously copied to this variable.
LOP	REAL	Read/write access Enables to assign the local output for the sequential control if the owner is the Program (Owner is 0) and the mode is MANUAL. Otherwise, the current output (OP output) is continuously copied to this variable.
PV	REAL	Read-only access Refer to the PV input pin, page 524.
Owner	BOOL	Read-only access Refer to the Owner input/output pin, page 526.
Man	BOOL	Write-only access 1 = Allows the controller to be changed to Manual mode if the owner is the Program. The DFB automatically sets the signal to 0 after processing the signal.
Auto	BOOL	Write-only access 1 = Allows the controller to be switched to Auto mode (if the owner is the Program. The DFB automatically sets the signal to 0 after processing the signal.
Mode	BOOL	Read-only access 1 = Auto 0 = Manual Refer to the PIDADVCTLGP_ST.CFGW.Mode input/output pin, page 526.
Rem	BOOL	Read/write access 1 = Allows the DFB to be configured for a remote set-point -RSP, if the owner is the Program and the mode is Auto. 0 = Allows the DFB to be configured for a local set-point -LSP, if the owner is the Program and the mode is Auto.
ILCKD	BOOL	Read-only access The signal evaluation depending on the ILCK input and the PIDADVCTLGP_ST.STW.ILCKBP input/output is shown in the following table.

ILCKD	ILCKBP	SC.ILCKD
OFF	-	OFF
ON	OFF	ON
ON	ON	OFF

Control – PIDMUXGP

For the Control of PIDMUXGP, refer to, page 529.

Control – PIDTUNEGP

For the Control of PIDTUNEGP, refer to, page 532.

Control – ADVCTLDATA

Overview

This section describes the ADVCTLDATA DFB.

Description

General Description

This ADVCTLDATA DFB is generated when $PIDADVGP template is linked with $AnalogOutputGP template either directly or via $AnalogSelectGP template.

This DFB takes the DownStream Object's information (Status, Value etc) and pass it to the Input-output variable. Template generates the same variable on ADVCTLDATA_IO pin of both PIDADVCTLGP DFB and ADVCTLDATA DFB, So the DownStream object information gets available in the PIDADVCTLGP DFB.

Function Description

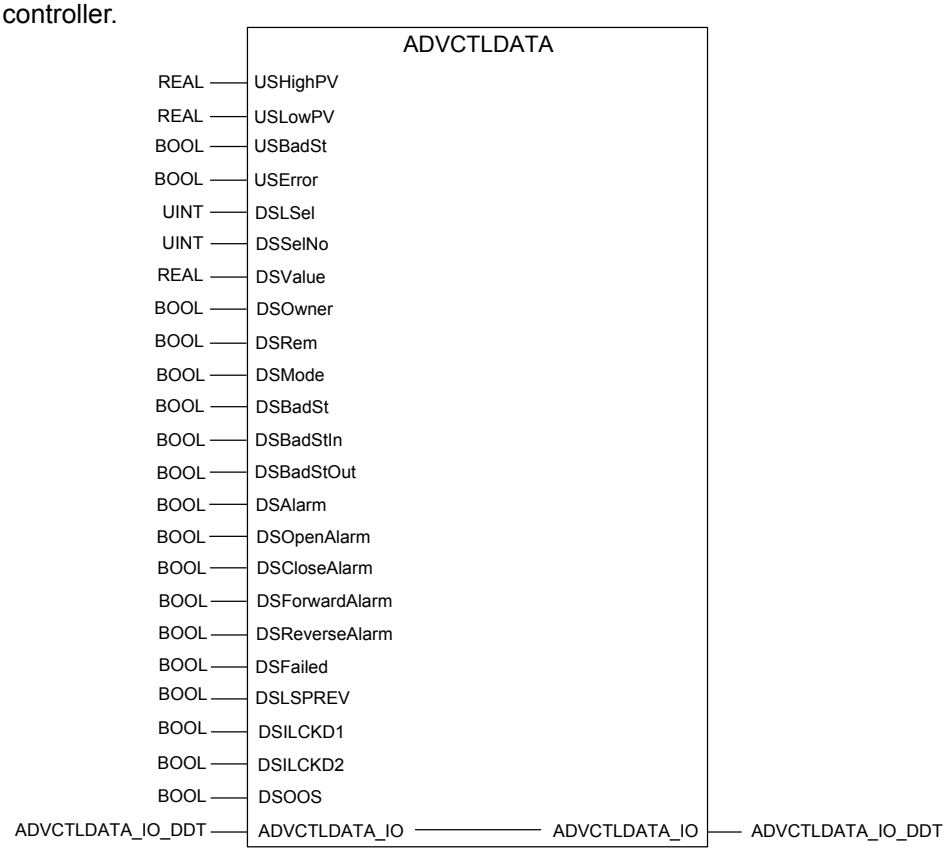
Function	Description
Upstream object in Bad State	When Upstream Object is in Bad State or in Error (USBadSt or USError =1), this information is passed to USBadSt variable of ADVCTLDATA_IOData structure.
DownStream object in Bad State	When DownStream Object is in Bad State (DSBadSt or DSBadStIn or DSBadStOut is active), this information is passed to DSBadSt variable of ADVCTLDATA_IO Data structure.
DownStream object in Interlocked State	When the DownStream objects are in interlocked state, this information is passed to DSILCKD variable of ADVCTLDATA_IO Data structure.
Output Track Enable	When Selected setpoint (of AnalogSelect block) is not same as input number on which this PID is linked in AnalogSelect block OR DownStream object is in Operator owner OR DownStream object is in Program Local OR DownStream object is in Interlocked state OR DownStream object is in Manual mode, the OPTrackEn variable of ADVCTLDATA_IO Data structure is set to 1. When OPTrackEn variable is 1 then PIDADVCTL output tracks the value coming from the DownStream.
DownStream Value	The output of the Analog output/Control Valve/Variable Speed Motor blocks are passed to DSValue variable of ADVCTLDATA_IO Data structure.
Upstream PV High range	Upstream blocks High PV range is passed to USHighPV variable of ADVCTLDATA_IO Data structure.
Upstream PV Low range	Upstream blocks Low PV range is passed to USLowPV variable of ADVCTLDATA_IO Data structure.

Function	Description
DownStream object Alarm State	The DownStream block alarm status is passed to `DSAlarm` variable of `ADVCTLDATA_IO` Data structure.
DownStream object detected failure State	The DownStream block detected Failure status is passed to `DSFailed` variable of `ADVCTLDATA_IO` Data structure.
DownStream object Out of Service State	The DownStream block `OutofService` status is passed to DSOOS variable of `ADVCTLDATA_IO` Data structure.

DFB Representation

Representation

This DFB has been specifically designed for use with the FBD language of the controller.



Inputs

Input Parameter Description

Parameter	Type	Description
`USHighPV`	`REAL`	Upstream process value high range.
`USLowPV`	`REAL`	Upstream process value low range.
`USBadST`	`BOOL`	Read–only 1 = Channel input detected failure.
`USError`	`BOOL`	1 = detected error active in `AnalogSelect` DFB.
`DSLSel`	`UINT`	Indicates which setpoint is selected as output in `AnalogSelect` DFB.

Parameter	Type	Description
DSLSelNo	UINT	AnalogSelect DFB input pin to which PIDDFB output is connected.
DSValue	REAL	Downstream DFB output value.
DSOwner	BOOL	Owner (0 = Program, 1 = Operator).
DSRem	BOOL	1 = Remote mode selected.
DSMode	BOOL	1 = Auto, 0 = Manual.
DSBadST	BOOL	1 = Output Channel associated with ChOut is not operational.
DSBadStIn	BOOL	1 = CValve AISignal detected failure.
DSBadStout	BOOL	1 = CValve AOSignal detected failure.
DSAlarm	BOOL	1 = Confirmation detected failure active.
DSOpenAlarm	BOOL	1 = Open confirmation alarm.
DSCloseAlarm	BOOL	1 = Close confirmation alarm.
DSForwardAlarm	BOOL	1 = Motor forward confirmation alarm.
DSReverseAlarm	BOOL	1 = Motor reverse confirmation alarm.
DSFailed	BOOL	1 = Detected failure active.
DSLSPREV	BOOL	1 = Forward, 0 = Reverse
DSILCKD1	BOOL	Interlocked 1
DSILCKD2	BOOL	Interlocked 2
DSOSS	BOOL	Out of service.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
ADVCTLDATA_IO_DDT	ADVCTLDATA_IO_DDT	Advance control data structure.

ADVCTLDATA_IO_DDT Type

Name	Bit Name	Bit	Type	Description
STW	–	–	WORD	Provides the device status usually used from the monitoring subsystem, and allows data to be kept in the memory. Read-only access to the data contained in this word.
	USBadSt	*Bit0*	BOOL	Upstream object channel detected failure.
	DSBadSt	*Bit1*	BOOL	Downstream object channel detected failure.
	OPTrack-En	*Bit2*	BOOL	1 = output tracking enable.
	DSAlarm	*Bit3*	BOOL	1 = downstream object alarm.
	DSFailed	*Bit4*	BOOL	1 = downstream object detected failed.
	DSILCKED	*Bit5*	BOOL	1 = downstream object interlocked.
	DSOSS	*Bit6*	BOOL	1 = downstream object out of service.
	DSOwner	*Bit7*	BOOL	1 = downstream owner is operator
TrackValue	–	–	REAL	Track value

Name	Bit Name	Bit	Type	Description
USHighPV	–	–	REAL	Upstream PV high range
USLowPV	–	–	REAL	Upstream PV low range

Supervision

Overview

This section describes the Supervision functions and runtime services that are available for the Advanced PID.

Genies

Genie Properties

Refer to Using Genies, page 62.

Genie Icons

At the beginning of the manual, you can find a description of the icons, page 63 that are part of the genies.

Representation

Refer to Representation, page 536, for the genies of the gpl_pidadv library:

Faceplate

Overview

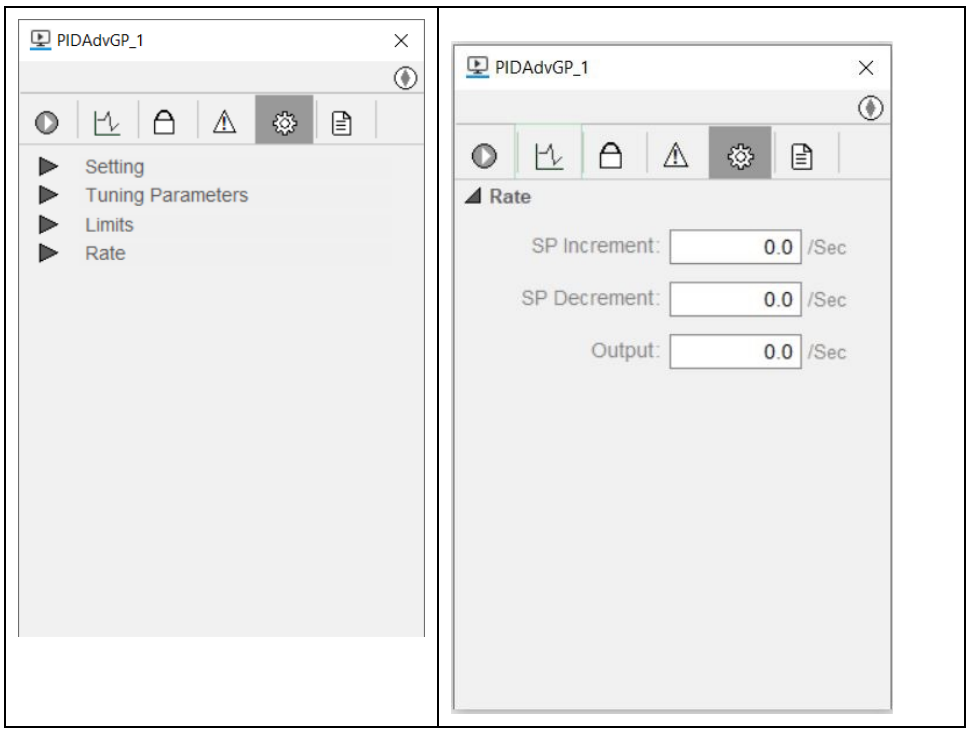
During operation, clicking a genie of the gpl_pidadv library allows you to display a faceplate with the following tabs:

- Tabs for core functions:
  - Operation, page 540
  - Engineering
  - Alarm, page 67
  - Event, page 69
  - Trend, page 69
- Tabs for optional functions, which appear only if configured:
  - Interlock, page 87
  - Failure, page 78
  - Measures, page 70

Engineering Tab

All the groups of an engineering tab except Rate are same as PID, refer Engineering Tab, page 541.

The figure shows an Rate group of an engineering tab:



The table describes the control function of `Rate` group of an engineering tab:

Label	Item name	Description	Security Parameter, page 520
SP Increment	SPIncRate	Setpoint increment rate	Set parameters
SP Decrement	SPDecRate	Setpoint decrement rate	
Output	OutRate	Output	

Items

Overview

This section describes the variables, equipstate variable, abnormal and action required conditions, alarms, events, and trends items of `Advanced PID`.

Variables

The table describes the variable items that are used by Supervision components:

Item name	Description	Enumeration	Address
Bit items derived from StatusWord.			
`ProgramSetpointMode`	Setpoint source for program owner	Remote/Sequence	*Bit0*
`InterlockActive`	Interlock status	Interlocked/Healthy	*Bit2*
`OverrideOutput`	PID override output selection	OnOff	*Bit3*
`AutoConfigurationInvalid`	Auto mode configuration is invalid	–	*Bit4*
`Warning`	PID alert is active	–	*Bit5*
`OutputLow`	Output is less than or equal to the low threshold of output low limit	–	*Bit6*
`OutputHigh`	Output is more than or equal to the high threshold of output high limit	–	*Bit7*

Item name	Description	Enumeration	Address
PVLimitIdentical	Lower limit of the PV range is identical to the upper limit of the PV range	–	*Bit8*
Bit items derived from PIDTuneStatusWord.			
PIDTuneCalculationError	PID tuning error in a floating point value calculation	–	*Bit0*
PIDTuneInvalidInput	Invalid value recorded at one of the floating point inputs		*Bit1*
PIDTuneDivisionByZero	Division by zero with calculation in floating point values		*Bit2*
PIDTuneCapacityOverflow	Capacity overflow during floating point value calculation		*Bit3*
PIDTunePerfOutOfRange	Parameter perf is outside the range		*Bit4*
PIDTuneThreshold	Thresholds pv_inf and pv_sup of the controller to be set		*Bit7*
PIDTuneParaCConnected	PARA_C output is not connected to the parameters of an autotunable controller		*Bit8*
PIDTuneFailed	Autotuning unsuccessful		*Bit9*
PIDTuneSuccess	Last autotune was successful		*Bit10*
Bit items derived from PIDTuneDiagnosisWord.			
PIDTuneRunning	Autotuning is running	Active/Inactive	*Bit0*
PIDTuneAborted	Autotuning aborted		*Bit1*
PIDTuneParError	Parameter error		*Bit2*
PIDTuneAlterPar	Alteration of parameters which have just been set automatically		*Bit3*
PIDTuneSysError	Stop as a consequence of system error		*Bit4*
PIDTuneSaturated	Process value saturated		*Bit5*
PIDTuneAlterSmall	Alteration too small		*Bit6*
PIDTuneSamplInvalid	Sampling interval invalid		*Bit7*
PIDTuneIncomReact	Incomprehensible reaction		*Bit8*
PIDTuneNonStabMeas	Non-stabilized measuring at the start		*Bit9*
PIDTuneTMaxShort	Length of actuating pulse Stellimpulses too short		*Bit10*
PIDTuneNoise	Too much noise/interference		*Bit11*
PIDTuneTMaxLong	Length of actuating pulse Stellimpulses too long		*Bit12*
PIDTuneExThresh	Process with significant exceeding of the thresholds		*Bit13*
PIDTuneMinPhase	Process without minimum phase		*Bit14*
PIDTuneAsymProcess	Asymmetric process		*Bit15*
Bit items derived from PIDTuneConfigurationWord.			
PIDTuneStart	Start autotune command	–	*Bit0*
PIDTuneStop	Stop autotune command		*Bit1*
PIDTuneTransfer	Transfer parameters		*Bit2*
PIDTuneRetPrevPar	Return previous parameters		*Bit3*
Bit items derived from PIDMuxConfigurationWord2.			
PIDMuxAction2	Mode of PID operation	–	*Bit3*
PIDMuxDerivativeApplied-To2	Output control action		*Bit4*
PIDMuxMixPar2	PID controller calculation formula is mixed or parallel		*Bit5*

Item name	Description	Enumeration	Address
Bit items derived from PIDMuxConfigurationWord1.			
PIDMuxAction1	Mode of PID operation	–	*Bit3*
PIDMuxDerivativeApplied-To1	Output control action		*Bit4*
PIDMuxMixPar1	PID controller calculation formula is mixed or parallel		*Bit5*
Bit items derived from PIDMuxConfigurationWord.			
PIDMuxOwnerSelect	Owner of control module	–	*Bit0*
PIDMuxParaSelect	Owner of control module		*Bit1*
Bit items derived from ConfigurationWord.			
OwnerSelect	Owner of control module	Operator/Program	*Bit0*
InterlockBypass	Interlock bypass command	Bypassed/Normal	*Bit1*
Mode	PID operating mode selection	Auto/Manual	*Bit2*
Action	Mode of PID operation	Direct/Reverse	*Bit3*
DerivativeAppliedTo	Output control action	Error/Present Value	*Bit4*
MixPar	PID controller calculation formula is mixed or parallel	Only Proportional/All Terms	*Bit5*
IO Items			
StatusWord	Status word	–	PID_ST.STW
ConfigurationWord	Configuration word		PID_ST.CFGW
PIDTuneDerTime	Derivative time	–	PID_PIDTuneDerTime
PIDTuneIntTime	Integral time	–	PID_PIDTuneIntTime
PIDTuneGain	Proportional gain	–	PID_PIDTuneGain
PIDTuneConfigurationWord	Configuration word bit array	–	PID_PIDTUNEGP_ST_CFGW
PIDTuneDiagnosisWord	Diagnosis word bit array	–	PID_PIDTUNEGP_ST_DIAGW
PIDTuneStatusWord	Status word bit array	–	PID_PIDTUNE_ST_STW
SetpointLowLimit	Define the minimum value of the setpoint	–	PID_SetpointLowLimit
SetpointHighLimit	Define the maximum value of the setpoint	–	PID_SetpointHighLimit
ActionRequired	Action required	–	PID_ActionRequired
Abnormal	Abnormal	–	PID_Abnormal
DerivativeGain	Derivative gain	–	PID_DerivativeGain
DerivativeTime	Derivative time	–	PID_DerivativeTime
IntegralTime	Integral time	–	PID_IntegralTime
ProportionalGain	Proportional gain	–	PID_Gain
OutputLowLimit	Define the minimum value of the output	–	PID_OutputLowLimit
OutputHighLimit	Define the maximum value of the output	–	PID_OutputHighLimit
Output	Output	–	PID_Output
Setpoint	Setpoint	–	PID_Setpoint
AutoSP	Setpoint	–	PID_AutoSP
SPIncRate	Setpoint increment rate	–	PID_SPIncRate
SPDecRate	Setpoint decrement rate	–	PID_SPDecRate
OutRate	Output rate	–	PID_OutRate

Item name	Description	Enumeration	Address
PresentValue	Present value of measurement	–	PID_PresentValue
PIDTunePrevDerTime	Autotune previous autotune derivative time	–	PID_ PIDTunePrevDerTime
PIDTunePrevIntTime	Autotune previous integration time	–	PID_ PIDTunePrevIntTime
PIDTunePrevGain	Autotune previous proportional constant	–	PID_PIDTunePrevGain
PIDTuneTime	Autotune time	–	PID_PIDTuneTuneTime
PIDTuneAmp	Step amplitude	–	PID_PIDTuneTuneAmp
PIDMuxOutputHighLimit1	Define the maximum value of the output	–	PIDMultiplexer_ OutputHighLimit1
PIDMuxOutputLowLimit1	Define the minimum value of the output	–	PIDMultiplexer_ OutputLowLimit1
PIDMuxProportionalGain1	Proportional gain	–	PIDMultiplexer_Gain1
PIDMuxIntegralTime1	Integral time	–	PIDMultiplexer_ IntegralTime1
PIDMuxDerivativeTime1	Derivative time	–	PIDMultiplexer_ DerivativeTime1
PIDMuxOutputHighLimit2	Define the maximum value of the output	–	PIDMultiplexer_ OutputHighLimit2
PIDMuxOutputLowLimit2	Define the minimum value of the output	–	PIDMultiplexer_ OutputLowLimit2
PIDMuxProportionalGain2	Proportional gain	–	PIDMultiplexer_Gain2
PIDMuxIntegralTime2	Integral time	–	PIDMultiplexer_ IntegralTime2
PIDMuxDerivativeTime2	Derivative time	–	PIDMultiplexer_ DerivativeTime2
PIDMuxConfigurationWord	Multiplexer configuration word bit array	–	PID_PIDMUX_ST_CFGW
PIDMuxConfigurationWord1	Multiplexer configuration word bit array	–	PID_PIDMUX_ST_CFGW1
PIDMuxConfigurationWord2	Multiplexer configuration word bit array	–	PID_PIDMUX_ST_CFGW2
PID_PIDMUX_ST_STW	Multiplexer status word bit array	–	PID_PIDMUX_ST_STW
PIDMuxPIDSelect	Parameter set selection	–	PID_PIDMuxPIDSelect
ORDsp	Optimal range display	–	TRUE
ORHigh	Optimal high	–	PID_PIDGP_ORHigh
ORLow	Optimal low	–	PID_PIDGP_ORLow
PR	Practical range	–	PID_PIDGP_PR
PRHigh	Practical range high	–	PID_PIDGP_PRHigh
PRLow	Practical range low	–	PID_PIDGP_PRLow
OptimalDevHigh	Optimal deviation high	–	PID_PIDGP_ OptimalDevHigh
OptimalDevLow	Optimal deviation low	–	PID_PIDGP_ OptimalDevLow
Calculated Items			
EquipState	Output information For more details, refer to EquipState calculated variable, page 966.	–	GPL_PIDEquipState
Owner	Current owner of the PID For more details, refer to calculated variable, page 58.		GPL_Owner
ModeSymbol	Current mode of the PID		GPL_PIDMode

Item name	Description	Enumeration	Address
	For more details, refer to calculated variable, page 59.		
Abnormal	Shows abnormal condition of the PID For more details, refer to abnormal conditions, page 966.		GPL_PIDAbnormal
ActionRequired	Shows action required condition of the PID For more details, refer to action required conditions, page 967.		GPL_PIDActionRequired

EquipState Calculated Variable

The table describes the *EquipState* calculated variable:

Equipment State	Conditions	Enumeration
Active with Interlock	ILCK x OP	On Interlocked
Passive with Interlock	ILCK x Not OP	Off Interlocked
Active	Not ILCK x OP	On
Passive	Not ILCK x Not OP	Off

Abnormal Conditions

The table describes the abnormal conditions:

Item name	Description	Address
Bit items derived from Abnormal variable word.		
AbnormalOwner	Abnormal owner status appears if the current owner is not a normal owner which configured at the instance level.	*Bit0*
AbnormalMode	Abnormal mode status appears if the current operating mode is not normal mode.	*Bit1*
AbnormalInterlockBy-pass	Abnormal interlock bypass condition appears if the Interlock bypass is enabled.	*Bit2*
AbnormalAction	Abnormal action	*Bit3*
AbnormalHighDeviatio-nEnable	If low alarm enable is active.	*Bit4*
AbnormalLowLowEnable	If low-low alarm enable is active.	*Bit5*
AbnormalLowEnable	If low alarm enable is active.	*Bit6*
AbnormalLowDeviatio-nEnable	If low alarm enable is active.	*Bit7*
AbnormalOperatorTab	Abnormal status of operator tab. This condition appears if the Interlock bypass is enabled.	*Bit8*
AbnormalHighHighEnable	If high-high alarm enable is active.	*Bit9*
AbnormalHighEnable	If high alarm enable is active.	*Bit10*
AbnormalInterlockTab	Abnormal status of interlock tab.	*Bit11*
AbnormalEngineeringTab	Abnormal status of engineering tab.	*Bit12*

Action Required Conditions

The table describes the action required conditions:

Item name	Description	Address
Bit items derived from ActionRequired variable word.		
`ActionInterlockTab`	Action required status of interlock tab.	*Bit11*

Alarms

Alternate Services:

- LAlarm, page 123
- DAlarm, page 128

Events

Event conditions are conditions which are detected during the operation of control module or the logging of operator actions:

Item name	Description
`StateAlarm`	All changes in the PID status and output.
`ModeAlarm`	All changes in the mode and owner.
`ActionAlarm`	All actions by the operator.

Trends

The table describes the trend tags that are managed by Supervision components and their expression:

Item name	Description	Expression
`PV`	Present value	`InstanceName_PIDADVGP_PV`
`SP`	Current setpoint	`InstanceName_PIDADVGP_SP`
`OP`	Current output	`InstanceName_PIDADVGP_OP`

Two Loop PID Control

What's in This Chapter

Template 968
Control 970
Supervision 970

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the `PID2LoopGP` with monitoring interface.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$PID2LoopGP` template.

This process template helps monitor and control the controller easily from the Supervision system and provides the operating modes used in the remaining EcoStruxure Process Expert process templates

Description

General Description

The `$PID2LoopGP` template defines operation of a typical 2-Loop PID. As a feedback controller, it delivers the control output at desired levels.

The `$PID2LoopGP` process control module template is used in a process where two PIDs are required either in individual control or in cascaded control. For example, you can use this template with crusher control for controlling the feed and load of the crusher.

Refer to the General Description, page 519 of PID.

Function Description

The `$PID2LoopGP` block has the following control functions:

- Loop1 control
- Loop2 control

Refer to the Functional Description, page 519 of PID.

Parameters

Parameters

The following table lists the system parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Loop 1 name	`String`	PID1	–

Section	Parameter description	Type	Default	Additional remarks
	Loop 2 name		PID2	–
	Cascade mode	`Boolean`	True	–

NOTE:

- The Loop name does not supports just the numeric characters.
- The character length should not exceed 6.

Control

For the control parameters refer to, page 520.

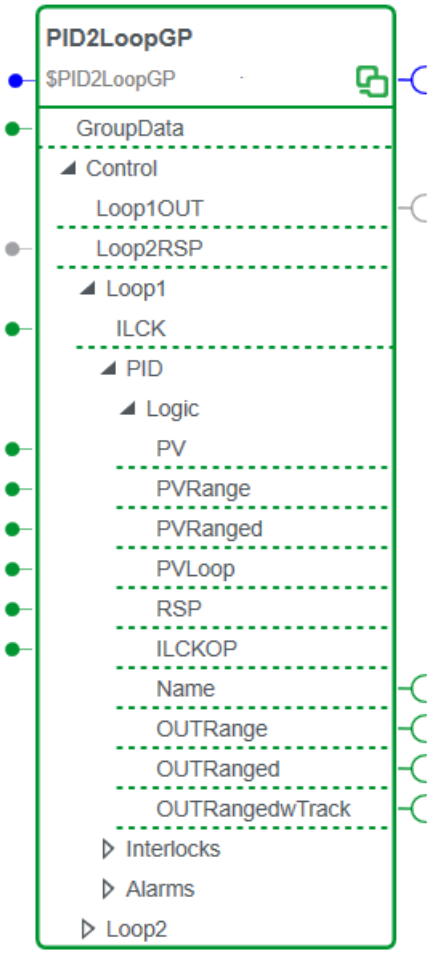
Supervision

For Supervision parameters refer to, page 520.

NOTE: It is mandatory to mention the display parameter **Process page name** .

Interfaces

This figure shows the `PID2LoopGP` template as it appears in `Links Editor` and an example of other templates connected to it:



NOTE:

- For analog input and PID faceplate navigation, it is recommended to define the parameter `Process Page Name` where these instances will be located in graphic page else during navigation the process display area will not show the content. For parameter description, refer to, page 43.
- The “Display page name” must be defined if 2LoopPID or 4LoopPID is used while connecting analog input else the display area will not show the content once the user opens the child page.

The following table lists the link interface that creates link symbol on faceplate:

Interface name	Runtime parameter
PV	PresentValueNavigate
PVRange	
PVRanged	
PVLoop	
RSP	SetpointNavigate
ILCKOP	
Name	
OUTRange	OutputNavigate
OUTRanged	
OUTRangedwTrack	

For more details, refer to Supervision chapter > Faceplate > Navigation Link of General Purpose Library User Guide.

Refer to common control module template interface, page 47 for more details on common interfaces.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

Overview

`PID2LoopGP` control can be used in a process where two PIDs are required either in individual control or in cascaded control.

For the Control of `PIDCTLGP` refer to, page 524.

For the Control of `PIDMUXGP` refer to, page 529.

For the Control of `PIDTUNEGP` refer to, page 532.

Supervision

Overview

This section describes the Supervision functions and runtime services that are available for the `PID2LoopGP`.

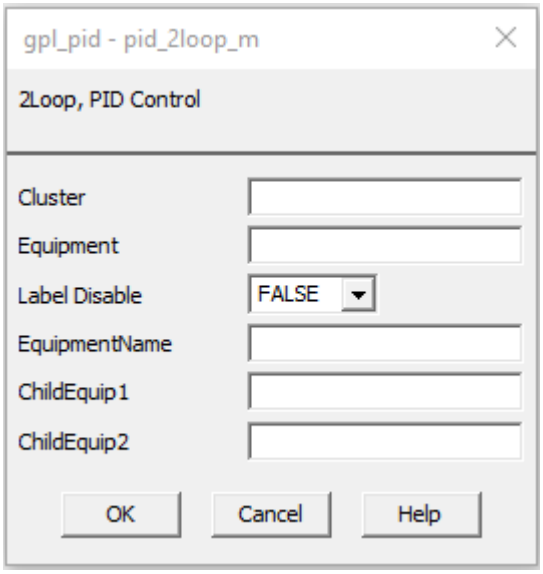
Genies

Genie Properties

Once you have assigned a genie to a page, double-click it to display its properties.

Genies of the GeneralPurposeLibrary project have the **Cluster** and **Equipment**, **Label Disable**, **Configurable Span**, **Default Span**, **Item**, and **Descriptor**, properties.

The figure shows an example of the properties dialog box of the `PID2Loop` genie:



The table describes the properties of the `PID2Loop` genie:

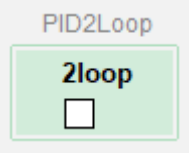
Genie properties	Description
Cluster	Cluster name of the equipment.
Equipment	Equipment name
Label Disable	By default, Label Disable is False, enter True if you need to hide the label of an object.
Equipment Name	Equipment name
ChildEquip1	Child equipment 1
ChildEquip2	Child equipment 2

Genie Icons

At the beginning of the manual, you can find a description of the icons, page 63 that are part of the genies.

Representation

The table describes the genies of the `PID2LoopGP` library:

Graphic Symbol	Genie name
	PID2Loop

Faceplate

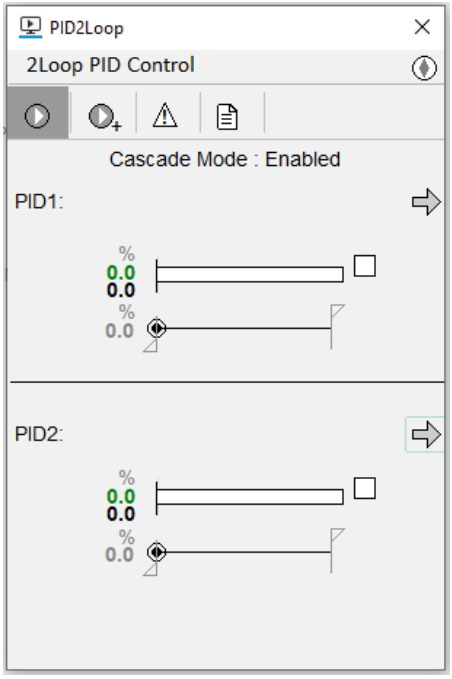
Overview

During operation, clicking a genie of the `PID2LoopGP` library allows you to display a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Extended Operator Tab
  - Alarm Tab
  - Event Tab

Operator Tab

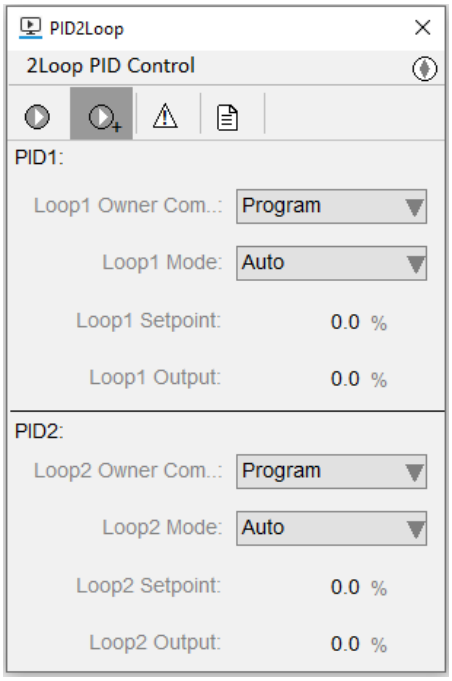
The figure shows an operator tab:



NOTE: Click on to open the PID faceplate. For more details on the PID faceplate tabs refer to the Faceplate section, page 540.

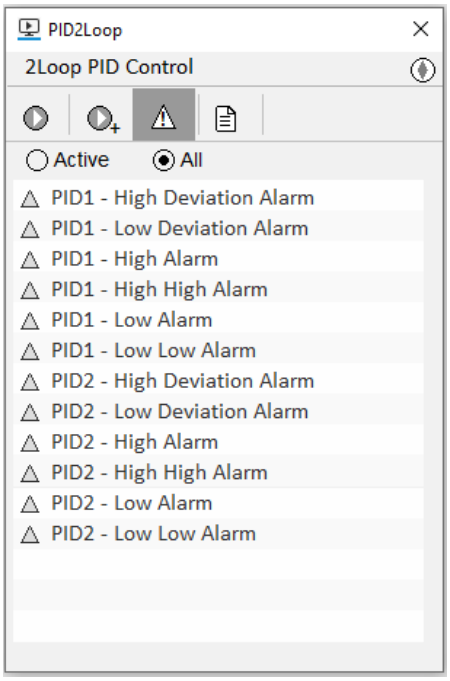
Extended Operator Tab

The figure shows an extended operator tab:



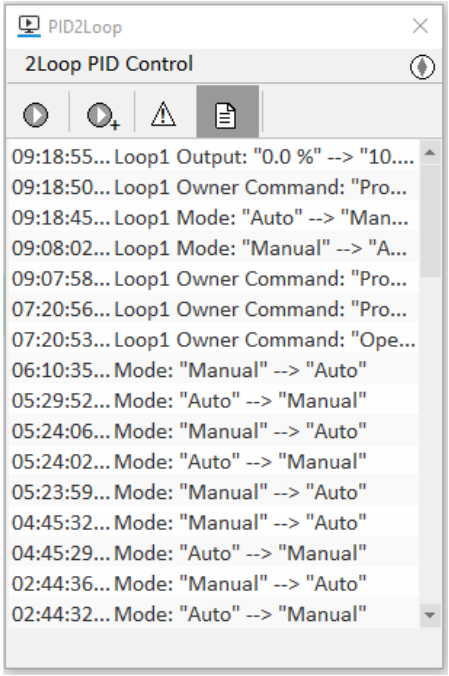
Alarm Tab

The figure shows the alarm tab:



Event Tab

The figure shows the event tab:



Items

Overview

Refer to the Items section, page 544 for details of the variables, equipstate variable, abnormal and action required conditions, alarms, events, and trends items of `PID2LoopGP`.

Four Loop PID Control

What's in This Chapter

Template 975
Control 978
Supervision 978

Overview

This chapter describes the functionality of template, control services, and Supervision functions of the `PID4LoopGP` with monitoring interface.

Template

Overview

This section describes the functionality, parameters, and interfaces of the `$PID4LoopGP` template.

This process template helps monitor and control the controller easily from the Supervision system and provides the operating modes used in the remaining EcoStruxure Process Expert process templates

Description

General Description

The `$PID4LoopGP` function block is used in a process where four PIDs are required either in individual control or in two cascaded control loops. For example, you can use this function block with crusher control, agitator control, and multi-process control.

For more details refer to the General Description, page 519 of PID.

Function Description

The Four Loop PID Control block is a generic control block. You can use this function block for a process where four PIDs are required either in individual control or in two cascaded control loops.

For example, you can use this control with crusher control, agitator control, and multi-process control.

The `$PID4LoopGP` block has the following control functions:

- Loop1 control
- Loop2 control
- Loop3 control
- Loop4 control

Parameters

Parameters

The following table lists the system parameters:

Section	Parameter description	Type	Default	Additional remarks
Configuration	Loop 1 name	`String`	PID1	–
	Loop 2 name		PID2	–
	Loop 3 Name		PID3	–
	Loop 4 Name		PID4	–
	Cascade Mode	`Enum`	Disabled (0)	• Disabled (0) • 2 Loop*(1) (1) • 2 Loop*(2) (2)

NOTE:

- The Loop name supports the numeric and alphabetic characters.
- The character length should not exceed 6.

Control

For the control parameters refer to, page 520.

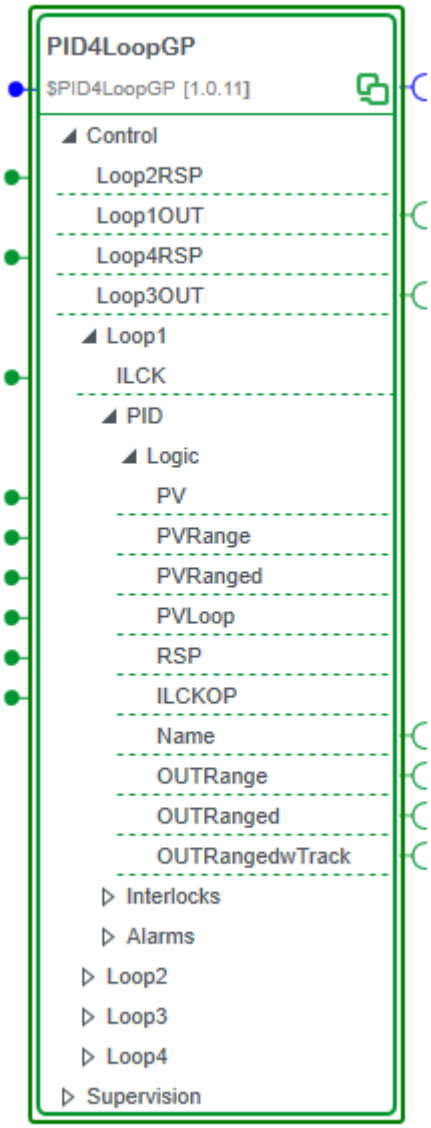
Supervision

For Supervision parameters refer to, page 520.

NOTE: It is mandatory to mention the display parameter **Process page name**.

Interfaces

This figure shows the `PID4LoopGP` template as it appears in `Links Editor`:



NOTE:

- For analog input and PID faceplate navigation, it is recommended to define the parameter `Process Page Name` where these instances will be located in graphic page else during navigation the process display area will not show the content. For parameter description, refer to, page 43.
- The “Display page name” must be defined if 2LoopPID or 4LoopPID is used while connecting analog input else the display area will not show the content once the user opens the child page.

The following table lists the link interface that creates link symbol on faceplate:

Interface identifier	Type/Role	Description
`Loop2RSP`	`$Real/Ref`	Edit Interface
`Loop1OUT`	`$Real/Def`	Edit Interface
`Loop4RSP`	`$Real/Ref`	Edit Interface
`Loop3OUT`	`$Real/Def`	Edit Interface
`ILCK`	`$Bool/Ref`	Edit Interface
`PV`	`$Real/Ref`	Edit Interface

Interface identifier	Type/Role	Description
PVRange	$Range/Ref	Edit Interface
PVRanged	$Rangedvalue/Ref	Edit Interface
RSP	$Real/Ref	Edit Interface
ILCKOP	$Real/Ref	Edit Interface
OUT Range	$Real/Def	Edit Interface
ILCK	$Bool/Ref	Edit Interface
OUT Ranged	$Rangedvalue/Def	Edit Interface
OUT Ranged/Track	$RangedTarck/Def	Edit Interface

For more details, refer to Supervision chapter > Faceplate > Navigation Link of General Purpose Library User Guide.

Refer to common control module template interface, page 47 for more details on common interfaces.

Refer to Group Control Link, page 135 to link the template with Group Control Link.

Control

Overview

PID4LoopGP control can be used in a process where four PIDs are required either in individual control or in cascaded control.

For the Control of PIDCTLGP refer to, page 524.

For the Control of PIDMUXGP refer to, page 529.

For the Control of PIDTUNEGP refer to, page 532.

Supervision

Overview

This section describes the Supervision functions and runtime services that are available for the PID4LoopGP.

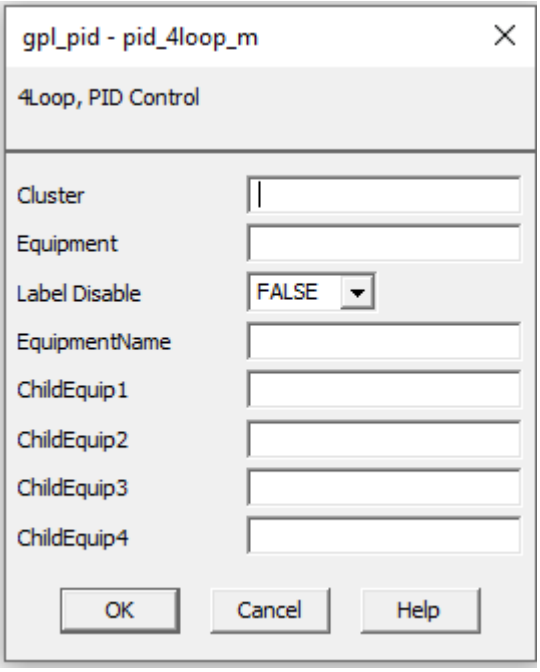
Genies

Genie Properties

Once you have assigned a genie to a page, double-click it to display its properties.

Genies of the GeneralPurposeLibrary project have the **Cluster** and **Equipment**, **Label Disable**, **Configurable Span**, **Default Span**, **Item**, and **Descriptor**, properties.

The figure shows an example of the properties dialog box of the `PID4Loop` genie:



The table describes the properties of the `PID4Loop` genie:

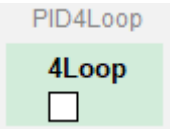
Genie properties	Description
Cluster	Cluster name of the equipment.
Equipment	Equipment name
Label Disable	By default, Label Disable is False, enter True if you need to hide the label of an object.
Equipment Name	Equipment name
ChildEquip1	Child equipment 1
ChildEquip2	Child equipment 2
ChildEquip3	Child equipment 3
ChildEquip4	Child equipment 4

Genie Icons

At the beginning of the manual, you can find a description of the icons, page 63 that are part of the genies.

Representation

The table describes the genies of the `PID4LoopGP` library:

Graphic Symbol	Genie name
	PID4Loop

Faceplate

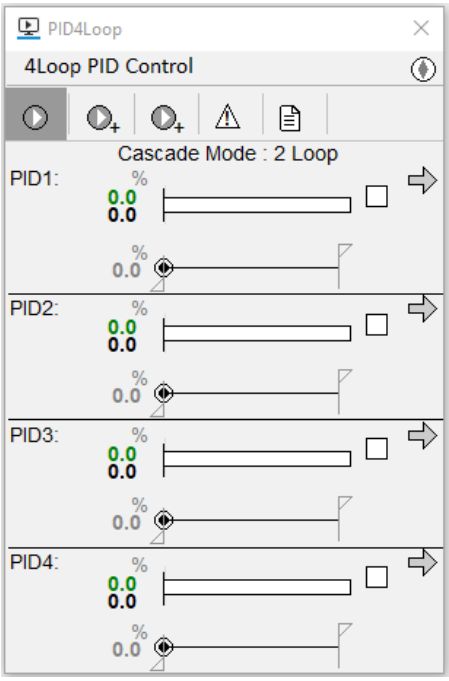
Overview

During operation, clicking a genie of the `PID4LoopGP` library allows you to display a faceplate with the following tabs:

- Tabs for core functions:
  - Operation
  - Extended Operator Tab
  - Alarm Tab
  - Event Tab

Operator Tab

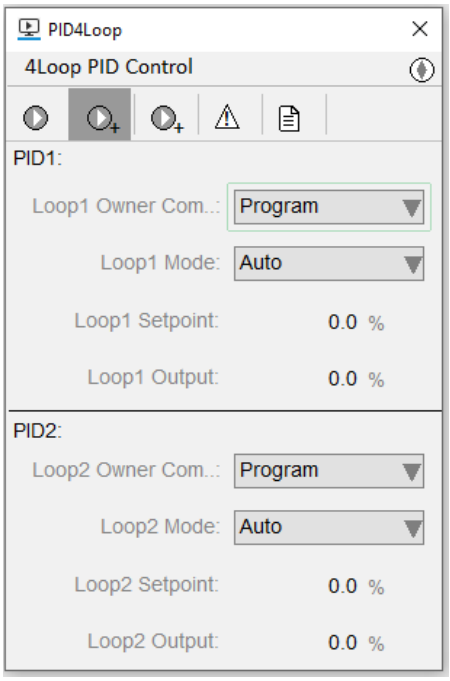
The figure shows an operator tab:

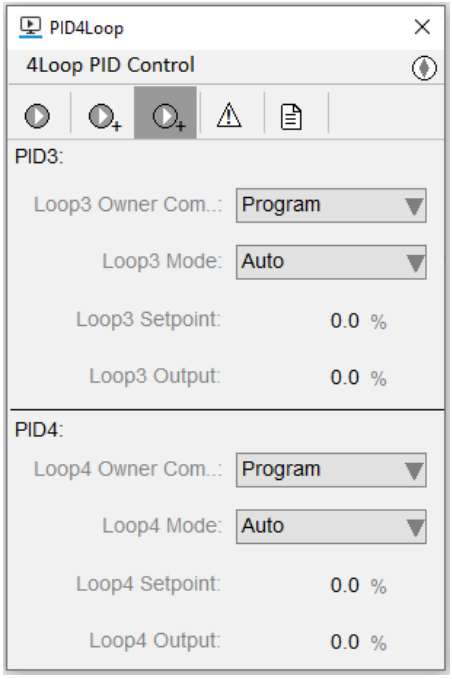


NOTE: Click on to open the PID faceplate. For more details on the PID faceplate tabs refer to the Faceplate section, page 540.

Extended Operator Tab

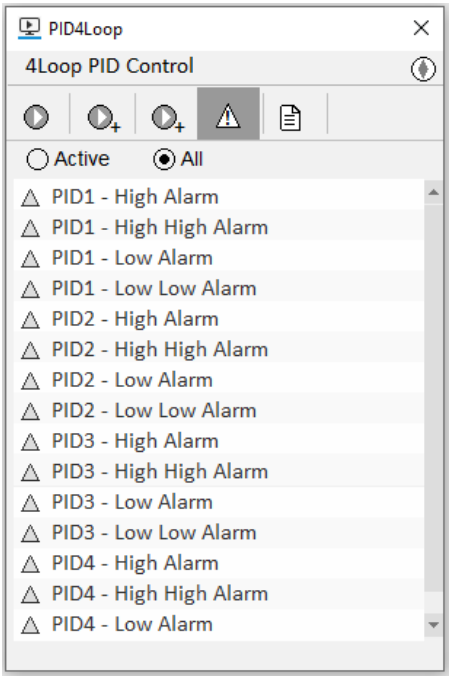
The figures below show the extended operator tabs:





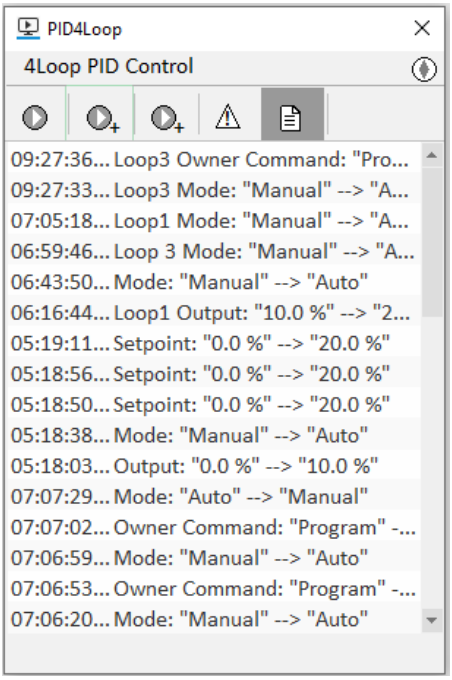
Alarm Tab

The figure shows the alarm tab:



Event Tab

The figure shows the event tab:



Items

Overview

Refer to the Items section, page 544 for details of the variables, equipstate variable, abnormal and action required conditions, alarms, events, and trends items of `PID4LoopGP`.

Index

$IMCGP
function description 617
general description 617
$LeadLagGP
function description 576
general description 576
$MAInpuGP
general description 462
$PhaseGP
template description 891
$PWMOutputGP
function description 638
general description 638
$RampGP
function description 564
$SequenceDFB
template description 657
$SplitRangeGP
general description 589
$Step3CtlGP
function description 602
general description 602

A

AALARM
function description 108
general description 108
AALARMGP
DFB representation 110
inputs 111
inputs/outputs 112
outputs 112
public variables 113
ACALCGP
inputs 757
access control
privilege levels 56
advance valve
supervision services 104
ADVCTLDATA
inputs 959
inputs/outputs 960
ADVCTLDATAR
DFB representation 558
inputs 558
inputs/outputs 558
alarm summary 722
Alarm tab
Faceplate Tabs 67
analog input 378
supervision services 385
analog output 395
supervision services 407
AOUTPUTLPGP
DFB representation 403
inputs 404
inputs/outputs 406
outputs 405
public variables 407
ARampGP
DFB representation 567
general description 564
inputs 567
inputs/outputs 569
outputs 568
public variables 570
ASELECTGP
DFB representation 704
inputs 704
inputs/outputs 705
outputs 705
public variables 706
auxiliary functions
$ASELECTGP 700
Analog Alarm 108
DVALVELPGP 309
MOTOR2LPGP 182
PHPARxxGP 937
SEQPARxxGP 693
UserPhasexxGP 942
USERSEQxxGP 697
auxiliary functions of equipment module
EMFAILEDGP 885
EMTRANSGP 883
auxiliary functions of PHASE
PHSTEPDESCGP 934
auxiliary functions of PHASEGP
PHFAILEDGP 932
PHSTABLESTEPGP 933
PHTRANSGP 930
auxiliary functions of sequential control
EMSTABLESTEPGP 886
EMSTEPDESCGP 887
SEQFAILEDGP 688
SEQSTABLESTEPGP 689
SEQSTEPDESCGP 690
SEQTRANSGP 686

B

BMXERT1604 516

C

Calculated
Variable 58
commands
phase control 898
sequential control 654, 842
CONDSUMGP
DFB representation 75
function description 74
general description 74
inputs 76
inputs/outputs 77
outputs 76
public variables 78
CVALVEGP
DFB representation 357
inputs 357
inputs/outputs 359
outputs 358
public variables 360
CValveLPGP
general description 362
CVALVELPGP
DFB representation 363
function description 362
inputs 363
inputs/outputs 366
outputs 365
public variables 367

D

DALARMGP
- DFB representation ... 126
- function description ... 124
- general description ... 124
- inputs ... 126
- inputs/outputs ... 127
- outputs ... 127
- public variables ... 128

DCALCGP
- DFB representation ... 752
- inputs ... 753
- outputs ... 753
- public variables ... 754

DEVMNTGP
- DFB representation ... 90
- function description ... 90
- general description ... 90
- inputs ... 90
- inputs/outputs ... 91
- outputs ... 91
- public variables ... 92

digital input ... 416
- supervision services ... 422

digital output ... 428
- supervision services ... 435

DINPUTGP
- DFB representation ... 419
- inputs ... 420
- inputs/outputs ... 421
- outputs ... 420
- public variables ... 421

DOUTPUTGP
- DFB representation ... 432
- inputs ... 432
- inputs/outputs ... 433
- outputs ... 432
- public variables ... 434

DVALVEGP
- DFB representation ... 300
- inputs ... 300
- inputs/outputs ... 305
- outputs ... 302
- public variables ... 306

DVALVELPGP
- DFB representation ... 309
- inputs ... 309
- inputs/outputs ... 311
- outputs ... 311
- public variables ... 312

E

EMCTL - example of use
- reusable sequence definition ... 871

EMCTL1 - example of use
- DFB representation ... 872
- reusable equipment module logic ... 872

EMCTL1-example of use
- general description ... 871

EMCTLGP
- description ... 855
- DFB representation ... 856
- inputs ... 856
- inputs/outputs ... 856
- public variables ... 859

EMFAILEDGP
- function call ... 885
- general description ... 885
- inputs/outputs ... 886

EMLOGICGP
- DFB representation ... 881
- inputs ... 882
- inputs/outputs ... 882

EMSTABLESTEPGP
- function call ... 886
- general description ... 886
- inputs/outputs ... 887

EMSTEPDESCGP
- description ... 887
- function calls ... 887
- inputs ... 888
- inputs/outputs ... 889
- public variables ... 889

EMTRANSGP
- function call ... 883
- general description ... 883
- inputs ... 884
- inputs/outputs ... 885

equipment module
- auxiliary functions of equipment module ... 883
- EMCTLGP ... 852
- EMCTLGP - example of use ... 871
- equipment module overview ... 839
- PHASEGP - example of use ... 944
- Strategy Execution Status controller ... 844

Equipment Module
- definition of statuses and commands ... 841
- overview ... 839
- Strategy Execution Status Matrix ... 843

equipment module control
- supervision services ... 862

Event tab
- Faceplate Tabs ... 69

F

faceplate
- description ... 65

faceplate tab
- icons ... 66

faceplate tabs
- Failures tab ... 78
- interlock tab ... 87
- measuress tab ... 70

Faceplate Tabs
- Alarm Tab ... 67
- Event Tab ... 69
- Trend Tab ... 69

Failures tab
- faceplate tabs ... 78

flow control ... 817
- supervision services ... 831

function block interface
- basic configuration ... 53
- continuous control ... 53
- illustration ... 54
- overview ... 52
- sequential control ... 53
- states and monitoring ... 53

G

general concepts
- function block interface ... 52
- modularity ... 52

General Description
function description 589

H

HVALVEGP
DFB representation 285
inputs 286
inputs/outputs 286
outputs 286
public variables 287
hyperlink services
Hyperlink services 50

I

ILCKONGP and ILCKOFFGP
DFB representation 83
function description 83
general description 82
ILCKONGP/ILCKOFFGP
inputs 83
inputs/outputs 85
outputs 84
public variables 86
IMCTLGP
DFB representation 621
inputs 622
inputs/outputs 623
outputs 623
public variables 625
interlock tab
faceplate tabs 87
Internal Model Controller
supervision services 626

L

LALARM
general description 118
LALARMGP
DFB representation 120
function description 118
inputs 120
inputs/outputs 121
outputs 121
public variables 122
LDLGCTL
inputs 580
LDLGCTLGP
DFB representation 580
outputs 580
public variables 582
Lead-Lag
supervision services 583
loadcell
supervision services 503

M

measuress tab
faceplate tabs 70
message box 712
supervision services 719
modularity
example 52
overview 52
motor maintenance
supervision services 97
MOTOR2GP
DFB representation 173
inputs 173, 178
outputs 175
public variables 179
MOTOR2LPGP
DFB representation 182
inputs 182
inputs/outputs 184
outputs 184
public variables 185
MotorGP
inputs/outputs 146
public variables 148
MOTORGP
DFB representation 143
input pin 143
output pin 144
MOTORLPGP
DFB representation 150
inputs 150
inputs/outputs 152
outputs 151
public variables 153
MOTORMNTGP
DFB representation 94
general description 94
inputs 95
inputs/outputs 95
outputs 95
public variables 96
Motors
MOTOR2GP 162
MOTORVSGP
DFB representation 203
inputs 203, 208
outputs 205
public variables 210
MOTORVSLPGP
DFB representation 213
function description 213
general description 212
inputs 214
inputs/outputs 216
outputs 215
public variables 217
MSGBOXGP
DFB representation 714
inputs 715
inputs/outputs 715
outputs 715
public variables 717
Multi Analog Input
supervision services 472
MVALVEDGP
DFB representation 329
inputs 329
inputs/outputs 334
outputs 331
public variables 335
MVALVEDLPGP
DFB representation 338
inputs 338
inputs/outputs 341
outputs 340
public variables 342
MVALVEGP

DFB representation 260
inputs 260
inputs/outputs 266
outputs 262
public variables 268
MVALVELPGP
DFB representation 270
inputs 270
inputs/outputs 273
outputs 272
public variables 274

P

Phase
auxiliary functions of PHASEGP 930
definition of statuses and commands 897
function description 896
operating modes 901
overview 896
PHASEGP 891
status controller 900
status matrix 899
supervision services 906
Phase control template
$PhaseGP 891
PHASEGP
DFB representation 901
inputs 901
inputs/outputs 902
PHASEGP - example of use
DFB representation 945
reusable phase definition 944
reusable phase logic 945
PHASEGP_1
inputs 942
inputs/outputs 943
PHASEGP-example of use
general description 944
PHFAILEDGP
function call 932
general description 932
inputs/outputs 933
PHPARxxGP
DFB representation 938–939
function description 937
general description 937
inputs 938
inputs/outputs 939
public variables 940
PHSTABLESTEPGP
function call 933
general description 933
inputs/outputs 934
PHSTEPDESCGP
description 934
function calls 934
inputs 935
inputs/outputs 936
public variables 936
PHTRANSGP
function call 930
general description 930
inputs 931
inputs/outputs 932
PID 519
PID2LoopGP 968
PID4LoopGP 975
PIDAdvCTLGP
inputs 954
outputs 955
PIDADVCTLGP
public variables 957
PIDADVGP 952
inputs/outputs 955
PIDCTLGP
DFB representation 524
inputs 524
inputs/outputs 526
outputs 525
public variables 528
PIDMUXGP
DFB representation 530
inputs 530
inputs/outputs 530
outputs 530
public variables 532
PIDTUNEGP
DFB representation 532
inputs/outputs 533
outputs 533
privileges
privilege levels for supervision actions 56
process templates
composition strategy 37
description 36
list of templates 32
naming convention 39
Pulse Width Modulated Controller
supervision services 645
pump set 785
supervision services 799
PWMCTLGP
DFB representation 641
inputs 642
inputs/outputs 643
outputs 642
public variables 644

R

RALARMGP
DFB representation 131
function description 130
general description 130
inputs 131
inputs/outputs 132
outputs 132
public variables 133
Ramp
supervision services 571
ratio controller 550, 558
RATIOCTLGP
DFB representation 554
inputs 554
inputs/outputs 555
outputs 555
public variables 556

S

SEQCTLGP
description 669
DFB representation 669
inputs/outputs 669
public variables 671
SEQCTLGP - example of use

DFB representation 681
reusable sequence definition 680
reusable sequence logic 681
SEQCTLGP-example of use
general description 680
SEQFAILEDGP
function call 688
general description 688
inputs/outputs 689
SEQPARxxGP
DFB representation 694
function description 693
general description 693
inputs 694
inputs/outputs 695
public variables 696
SEQSTABLESTEPGP
function call 689
general description 689
inputs/outputs 690
SEQSTEPDESCGP
description 690
function calls 690
inputs 691
inputs/outputs 692
public variables 692
SEQTRANSGP
function call 686
general description 686
inputs 687
inputs/outputs 688
sequential control 674
auxiliary functions of sequential control 686
definition of statuses and commands 653
function description 652, 839
operating modes 656, 844
overview 652
SEQCTLGP 662
SEQCTLGP - example of use 680
sequential control overview 652
status controller 656
status matrix 655
sequential control template
$SequenceDFB Example Use Cases 657
signal processing
ACALCGP 755
ALINEARGP 761
DCALCGP 751
Split-range
supervision services 596
SPLRGCTLGP
DFB representation 592
inputs 593
inputs/outputs 594
outputs 593
public variables 595
statuses
equipment module 841
Phase 897
sequential control 653
Step3 Controller
supervision services 610
STEP3CTLGP 607
DFB representation 606
inputs 606
inputs/outputs 608
public variables 609
supervision services
advance valve management 104
alarm summary management 722
analog input management 378, 385
analog output management 395, 407
BMXERT1604 516
digital input management 416, 422
digital output management 428, 435
equipment module control 862
flow control 817, 831
Internal Model Controller 626
Lead-Lag controller management 583
loadcell management 503
message box management 712, 719
motor maintenance management 97
Multi Analog Input 472
PhaseGP 906
PID 519
PID2LoopGP 968
PID4LoopGP 975
Pulse Width Modulated Controller 645
pump set 785, 799
Ramp management 571
ratio controller 550, 558
sequential control 674
Split-range management 596
Step3 Controller/positioner 610
Totalizer management 455

T

tabs
Alarm Tab 67
Event Tab 69
Trend Tab 69
Template
PIDADVGP 952
templates
list of process templates 32
Totalizer
supervision services 455
Trend tab
Faceplate Tabs 69

U

UserPhasexxGP
DFB representation 942
USERSEQxxGP
DFB representation 698
function description 697
general description 697
inputs 698
inputs/outputs 698

V

valve
Valve 229
Valve
hand valve 283
ValveGP
inputs/outputs 237
public variables 239
VALVEGP
DFB representation 234
input pin 234
output pin 235
VALVELPGP
DFB representation 241

inputs 241
inputs/outputs 243
outputs 242
public variables 244
VALVEMNTGP
DFB representation 102
general description 101
inputs 102
inputs/outputs 102
outputs 102
public variables 103
Valves
control valve 352
motorized control valve 254
motorized valve 324

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

As standards, specifications, and design change from time to time, please ask for confirmation of the information given in this publication.

© 2023 Schneider Electric. All rights reserved.

EIO0000004043.07