



Programmable Controller

**MELSEC iQ-R**  
series

MELSEC iQ-R Programming Manual  
(Process Control Function Blocks/Instructions)

---



# SAFETY PRECAUTIONS

---

(Read these precautions before using this product.)

Before using MELSEC iQ-R series programmable controllers, please read the manuals for the product and the relevant manuals introduced in those manuals carefully, and pay full attention to safety to handle the product correctly.

Make sure that the end users read this manual and then keep the manual in a safe place for future reference.

## CONDITIONS OF USE FOR THE PRODUCT

---

(1) Mitsubishi programmable controller ("the PRODUCT") shall be used in conditions;

i) where any problem, fault or failure occurring in the PRODUCT, if any, shall not lead to any major or serious accident; and

ii) where the backup and fail-safe function are systematically or automatically provided outside of the PRODUCT for the case of any problem, fault or failure occurring in the PRODUCT.

(2) The PRODUCT has been designed and manufactured for the purpose of being used in general industries.

MITSUBISHI SHALL HAVE NO RESPONSIBILITY OR LIABILITY (INCLUDING, BUT NOT LIMITED TO ANY AND ALL RESPONSIBILITY OR LIABILITY BASED ON CONTRACT, WARRANTY, TORT, PRODUCT LIABILITY) FOR ANY INJURY OR DEATH TO PERSONS OR LOSS OR DAMAGE TO PROPERTY CAUSED BY the PRODUCT THAT ARE OPERATED OR USED IN APPLICATION NOT INTENDED OR EXCLUDED BY INSTRUCTIONS, PRECAUTIONS, OR WARNING CONTAINED IN MITSUBISHI'S USER, INSTRUCTION AND/OR SAFETY MANUALS, TECHNICAL BULLETINS AND GUIDELINES FOR the PRODUCT.

("Prohibited Application")

Prohibited Applications include, but not limited to, the use of the PRODUCT in;

- Nuclear Power Plants and any other power plants operated by Power companies, and/or any other cases in which the public could be affected if any problem or fault occurs in the PRODUCT.
- Railway companies or Public service purposes, and/or any other cases in which establishment of a special quality assurance system is required by the Purchaser or End User.
- Aircraft or Aerospace, Medical applications, Train equipment, transport equipment such as Elevator and Escalator, Incineration and Fuel devices, Vehicles, Manned transportation, Equipment for Recreation and Amusement, and Safety devices, handling of Nuclear or Hazardous Materials or Chemicals, Mining and Drilling, and/or other applications where there is a significant risk of injury to the public or property.

Notwithstanding the above restrictions, Mitsubishi may in its sole discretion, authorize use of the PRODUCT in one or more of the Prohibited Applications, provided that the usage of the PRODUCT is limited only for the specific applications agreed to by Mitsubishi and provided further that no special quality assurance or fail-safe, redundant or other safety features which exceed the general specifications of the PRODUCTS are required. For details, please contact the Mitsubishi representative in your region.

(3) Mitsubishi shall have no responsibility or liability for any problems involving programmable controller trouble and system trouble caused by DoS attacks, unauthorized access, computer viruses, and other cyberattacks.

- For SIL2 Process CPUs

- (1) Although Mitsubishi Electric has obtained the certification for Product's compliance to the international safety standards IEC61508, IEC61511 from TUV Rheinland, this fact does not guarantee that Product will be free from any malfunction or failure. The user of this Product shall comply with any and all applicable safety standard, regulation or law and take appropriate safety measures for the system in which the Product is installed or used and shall take the second or third safety measures other than the Product. Mitsubishi Electric is not liable for damages that could have been prevented by compliance with any applicable safety standard, regulation or law.
- (2) Mitsubishi Electric prohibits the use of Products with or in any application involving, and Mitsubishi Electric shall not be liable for a default, a liability for defect warranty, a quality assurance, negligence or other tort and a product liability in these applications.
  - (a) power plants,
  - (b) trains, railway systems, airplanes, airline operations, other transportation systems,
  - (c) hospitals, medical care, dialysis and life support facilities or equipment,
  - (d) amusement equipments,
  - (e) incineration and fuel devices,
  - (f) handling of nuclear or hazardous materials or chemicals,
  - (g) mining and drilling,
  - (h) and other applications where the level of risk to human life, health or property are elevated.
- (3) Mitsubishi Electric shall have no responsibility or liability for any problems involving programmable controller trouble and system trouble caused by DoS attacks, unauthorized access, computer viruses, and other cyberattacks.

# INTRODUCTION

---

Thank you for purchasing the Mitsubishi Electric MELSEC iQ-R series programmable controllers.

This manual describes the procedures, details, and tag data of the process control function blocks.

This manual also describes the process control instructions that are used in the process control function blocks.

Before using this product, please read this manual and the relevant manuals carefully and develop familiarity with the functions and performance of the MELSEC iQ-R series programmable controller to handle the product correctly.


When applying the program examples provided in this manual to an actual system, ensure the applicability and confirm that it will not cause system control problems.

Please make sure that the end users read this manual.

---

## Point

- For details on the function blocks and FBD/LD language, refer to the following.

 MELSEC iQ-R Programming Manual (Program Design)

- For details on the faceplate, refer to the following.

 GX Works3 Operating Manual

 PX Developer Version 1 Operating Manual (Monitor Tool)

---

# CONTENTS

---

SAFETY PRECAUTIONS .....	1
CONDITIONS OF USE FOR THE PRODUCT .....	1
INTRODUCTION .....	3
RELEVANT MANUALS .....	13
TERMS .....	14
GENERIC TERMS AND ABBREVIATIONS .....	14
MANUAL PAGE ORGANIZATION .....	15

## PART 1 OVERVIEW

---

### CHAPTER 1 PROCESS CONTROL FUNCTION BLOCKS AND PROCESS CONTROL INSTRUCTIONS 20

---

1.1 Process Control Function Block .....	21
1.2 Process Control Instructions .....	22

### CHAPTER 2 PROCESS CONTROL FUNCTION BLOCK 23

---

2.1 Process Control Function Block Types .....	23
2.2 Program Supporting Process Control Function Blocks .....	24
2.3 Tag FB .....	25
Tag FB categories .....	26
Initial values for tag data and operation constants (public variables) .....	27
Control modes .....	27
I/O modes .....	28
2.4 User-defined Tag FB and Tag Access FB .....	29
Tag type list .....	30
2.5 Program Execution Control .....	31
Timer execution .....	31
Interrupt execution (fixed scan) .....	37
When a safety program is used .....	37
When a fixed scan execution type program/an interrupt program is used .....	37

### CHAPTER 3 PROCEDURE BEFORE USING PROCESS CONTROL FUNCTION BLOCK 38

---

3.1 Overview .....	38
3.2 Setting CPU Parameters .....	39
File Register Setting .....	39
Index Register Setting .....	40
Label Initial Value Reflection Setting .....	40
Tracking Setting .....	40
3.3 Declaring a tag FB (Tag FB Setting) .....	41
3.4 Editing an FBD/LD Program for Process Control .....	42
Creating a user-defined tag FB .....	45
3.5 Setting Initial Values of FBs (FB Property) .....	47
3.6 Converting and Writing .....	47
3.7 Checking FB Operation (Faceplate) .....	48
3.8 Troubleshooting .....	48

## PART 2 LISTS OF PROCESS CONTROL FUNCTION BLOCK

<b>CHAPTER 4</b>	<b>LISTS OF PROCESS CONTROL FUNCTION BLOCK</b>	<b>52</b>
4.1	General Process FBs .....	52
4.2	Tag Access FBs .....	59
4.3	Tag FBs .....	64

## PART 3 GENERAL PROCESS FB

<b>CHAPTER 5</b>	<b>ANALOG VALUE SELECTION AND AVERAGE VALUE</b>	<b>72</b>
5.1	High Selector (M+P_HS(_E)) .....	72
5.2	Low Selector (M+P_LS(_E)) .....	74
5.3	Middle Value Selection (M+P_MID(_E)) .....	76
5.4	Average Value (M+P_AVE(_E)) .....	78
5.5	Absolute Value (M+P_ABS(_E)) .....	80

<b>CHAPTER 6</b>	<b>CORRECTION OPERATION</b>	<b>82</b>
6.1	Function Generator (M+P_FG) .....	82
6.2	Inverse Function Generator (M+P_IFG) .....	85
6.3	Standard Filter (moving average) (M+P_FLT) .....	88
6.4	Engineering Value Conversion (M+P_ENG) .....	90
6.5	Engineering Value Inverse Conversion (M+P_IENG) .....	92
6.6	Temperature/Pressure Correction (M+P_TPC) .....	94
6.7	Summation (M+P_SUM) .....	96
6.8	Summation (Internal Integer Integration) (M+P_SUM2_) .....	98
6.9	Range Conversion (M+P_RANGE_) .....	101

<b>CHAPTER 7</b>	<b>ARITHMETIC OPERATION</b>	<b>103</b>
7.1	Addition (with coefficient) (M+P_ADD) .....	103
7.2	Subtraction (with coefficient) (M+P_SUB) .....	105
7.3	Multiplication (with coefficient) (M+P_MUL) .....	107
7.4	Division (with coefficient) (M+P_DIV) .....	109
7.5	Square Root (with coefficient) (M+P_SQR) .....	111

<b>CHAPTER 8</b>	<b>COMPARISON OPERATION</b>	<b>113</b>
8.1	Comparison (>) with setting value (M+P_GT) .....	113
8.2	Comparison (<) with setting value (M+P_LT) .....	115
8.3	Comparison (=) with setting value (M+P_EQ) .....	117
8.4	Comparison (>=) with setting value (M+P_GE) .....	119
8.5	Comparison (<=) with setting value (M+P_LE) .....	121

<b>CHAPTER 9</b>	<b>CONTROL OPERATION</b>	<b>123</b>
9.1	Lead-Lag (M+P_LLAG) .....	123
9.2	Integration (M+P_I) .....	126
9.3	Derivative (M+P_D) .....	129
9.4	Dead Time (M+P_DED) .....	131
9.5	High/Low Limiter (M+P_LIMT) .....	134

9.6	Variation Rate Limiter 1 (M+P_VLMT1) .....	137
9.7	Variation Rate Limiter 2 (M+P_VLMT2) .....	140
9.8	Dead Band (M+P_DBND) .....	143
9.9	Bump-less Transfer (M+P_BUMP) .....	145
9.10	Analog Memory (M+P_AMR) .....	147
9.11	8 Points Time Proportional Output (M+P_DUTY_8PT_).....	149

---

## **CHAPTER 10 BISTABLE SWITCH** **155**

10.1	Bistable Switch (Set Dominant) (M+SR) .....	155
10.2	Bistable Switch (Reset Dominant) (M+RS) .....	157
10.3	Latch FB (BOOL Type) (M+LATCH_BOOL) .....	159
10.4	Latch FB (REAL Type) (M+LATCH_REAL).....	160
10.5	Latch FB (WORD Type) (M+LATCH_REAL).....	161
10.6	Latch FB (DWORD Type) (M+LATCH_DWORD) .....	162

---

## **CHAPTER 11 EDGE DETECTOR** **163**

11.1	Rising Edge Detection (M+R_TRIG).....	163
11.2	Falling Edge Detection (M+F_TRIG) .....	164
11.3	Edge Detector (M+EDGE_CHECK).....	165

---

## **CHAPTER 12 COUNTER** **166**

12.1	Up Counter (M+CTU).....	166
12.2	Down Counter (M+CTD) .....	168
12.3	Up/Down Counter (M+CTUD) .....	169

---

## **CHAPTER 13 TIMER** **171**

13.1	Pulse Timer (High-speed Timer Type) (M+TP_HIGH) .....	171
13.2	Pulse Timer (Low-speed Timer Type) (M+TP_LOW).....	173
13.3	ON Delay Timer (High-speed Timer Type) (M+TON_HIGH) .....	175
13.4	ON Delay Timer (Low-speed Timer Type) (M+TON_LOW).....	177
13.5	OFF Delay Timer (High-speed Timer Type) (M+TOF_HIGH).....	179
13.6	OFF Delay Timer (Low-speed Timer Type) (M+TOF_LOW) .....	181

# **PART 4 TAG ACCESS FB**

---



---

## **CHAPTER 14 I/O CONTROL** **184**

14.1	Analog Input Processing (M+P_IN) .....	184
14.2	Output Processing-1 with Mode Switching (With Input Addition) (M+P_OUT1) .....	189
14.3	Output Processing-2 with Mode Switching (Without Input Addition) (M+P_OUT2) .....	194
14.4	Output Processing-3 with Mode Switching (With Input Addition and Compensation) (M+P_OUT3_) .....	198
14.5	Manual Output (M+P_MOUT) .....	206
14.6	Time Proportioning Output (M+P_DUTY) .....	208
14.7	Pulse Integrator (M+P_PSUM) .....	213
14.8	Batch Counter (M+P_BC) .....	216
14.9	Manual Setter (M+P_MSET_) .....	219

---

## **CHAPTER 15 LOOP CONTROL OPERATION** **223**

15.1	Ratio Control (Enable Tracking for primary loop) (M+P_R_T) .....	223
------	--	-----



15.2	Ratio Control (Disable Tracking for primary loop) (M+P_R)	227
15.3	Velocity Type PID Control (Enable Tracking for primary loop) (M+P_PID_T)	230
15.4	Velocity Type PID Control (Disable Tracking for primary loop) (M+P_PID)	236
15.5	2-degree-of-freedom PID Control (Enable Tracking for primary loop) (M+P_2PID_T)	242
15.6	2-degree-of-freedom PID Control (Disable Tracking for primary loop) (M+P_2PID)	248
15.7	2-degree-of-freedom Advanced PID Control (Enable Tracking for primary loop) (M+P_2PIDH_T)	254
15.8	2-degree-of-freedom Advanced PID Control (Disable Tracking for primary loop) (M+P_2PIDH)	262
15.9	Position Type PID Control (Enable Tracking for primary loop/Disable Tracking from secondary loop) (M+P_PIDP_T)	269
15.10	Position Type PID Control (Disable Tracking for primary loop/Disable Tracking from secondary loop) (M+P_PIDP)	276
15.11	Position Type PID Control (Enable Tracking for primary loop/Enable Tracking from secondary loop) (M+P_PIDP_EX_T)	283
15.12	Position Type PID Control (Disable Tracking for primary loop/Enable Tracking from secondary loop) (M+P_PIDP_EX)	291
15.13	Sample PI Control (Enable Tracking for primary loop) (M+P_SPI_T)	299
15.14	Sample PI Control (Disable Tracking for primary loop) (M+P_SPI)	305
15.15	I-PD Control (Enable Tracking for primary loop) (M+P_IPD_T)	310
15.16	I-PD Control (Disable Tracking for primary loop) (M+P_IPD)	316
15.17	Blend PI Control (Enable Tracking for primary loop) (M+P_BPI_T)	321
15.18	Blend PI Control (Disable Tracking for primary loop) (M+P_BPI)	326
15.19	High/Low Limit Alarm Check (M+P_PHPL)	331
15.20	2 position ON/OFF Control (Enable Tracking for primary loop) (M+P_ONF2_T)	335
15.21	2 position ON/OFF Control (Disable Tracking for primary loop) (M+P_ONF2)	338
15.22	3 position ON/OFF Control (Enable Tracking for primary loop) (M+P_ONF3_T)	341
15.23	3 position ON/OFF Control (Disable Tracking for primary loop) (M+P_ONF3)	345
15.24	Program Setter (M+P_PGS)	348
15.25	Multi-Point Program Setter (M+P_PGS2)	352
15.26	Loop Selector (Disable Tracking for primary loop) (M+P_SEL)	363
15.27	Loop Selector (Enable Tracking for primary loop only by CASIN_T) (M+P_SEL_T1)	366
15.28	Loop Selector (Enable Tracking for primary loop by CASIN_T1/_T2) (M+P_SEL_T2)	370
15.29	Loop Selector (Enable Tracking from secondary loop to primary loop) (M+P_SEL_T3)	374

**CHAPTER 16 TAG SPECIAL 378**

16.1	Change Control Mode (M+P_MCHG)	378
16.2	Primary Loop Control Mode Switching (M+P_MCHGPRMRY)	381

**PART 5 TAG FB**

**CHAPTER 17 LOOP TAG 386**

17.1	Velocity Type PID Control (Enable Tracking for primary loop) (M+M_PID_T)	386
17.2	Velocity Type PID Control (Disable Tracking for primary loop) (M+M_PID)	390
17.3	Velocity Type PID Control and DUTY Output (Enable Tracking for primary loop) (M+M_PID_DUTY_T)	393
17.4	Velocity Type PID Control and DUTY Output (Disable Tracking for primary loop) (M+M_PID_DUTY)	396
17.5	2-degree-of-freedom PID Control (Enable Tracking for primary loop) (M+M_2PID_T)	399
17.6	2-degree-of-freedom PID Control (Disable Tracking for primary loop) (M+M_2PID)	402
17.7	2-degree-of-freedom PID Control and DUTY O/P (Enable Tracking for primary loop) (M+M_2PID_DUTY_T)	405

17.8	2-degree-of-freedom PID Control and DUTY O/P (Disable Tracking for primary loop) (M+M_2PID_DUTY) . . . . .	408
17.9	2-degree-of-freedom Advanced PID Control (Enable Tracking for primary loop) (M+M_2PIDH_T_) . . . .	411
17.10	2-degree-of-freedom Advanced PID Control (Disable Tracking for primary loop) (M+M_2PIDH_) . . . .	417
17.11	Position Type PID Control (Enable Tracking for primary loop/Disable Tracking from secondary loop) (M+M_PIDP_T) . . . . .	424
17.12	Position Type PID Control (Disable Tracking for primary loop/Disable Tracking from secondary loop) (M+M_PIDP_) . . . . .	427
17.13	Position Type PID Control (Enable Tracking for primary loop/Enable Tracking from secondary loop) (M+M_PIDP_EX_T_) . . . . .	430
17.14	Position Type PID Control (Disable Tracking for primary loop/Enable Tracking from secondary loop) (M+M_PIDP_EX_) . . . . .	433
17.15	Sample PI Control (Enable Tracking for primary loop) (M+M_SPI_T) . . . . .	436
17.16	Sample PI Control (Disable Tracking for primary loop) (M+M_SPI) . . . . .	439
17.17	I-PD Control (Enable Tracking for primary loop) (M+M_IPD_T) . . . . .	442
17.18	I-PD Control (Disable Tracking for primary loop) (M+M_IPD) . . . . .	445
17.19	Blend PI Control (Enable Tracking for primary loop) (M+M_BPI_T) . . . . .	448
17.20	Blend PI Control (Disable Tracking for primary loop) (M+M_BPI) . . . . .	451
17.21	Ratio Control (Enable Tracking for primary loop) (M+M_R_T) . . . . .	454
17.22	Ratio Control (Disable Tracking for primary loop) (M+M_R) . . . . .	457
17.23	2 position ON/OFF Control (Enable Tracking for primary loop) (M+M_ONF2_T) . . . . .	460
17.24	2 position ON/OFF Control (Disable Tracking for primary loop) (M+M_ONF2) . . . . .	463
17.25	3 position ON/OFF Control (Enable Tracking for primary loop) (M+M_ONF3_T) . . . . .	466
17.26	3 position ON/OFF Control (Disable Tracking for primary loop) (M+M_ONF3) . . . . .	469
17.27	Monitor (M+M_MONI) . . . . .	472
17.28	Manual Output with Monitor (M+M_MWM) . . . . .	474
17.29	Batch Preparation (M+M_BC) . . . . .	476
17.30	Pulse Integrator (M+M_PSUM) . . . . .	478
17.31	Loop Selector (Disable Tracking for primary loop) (M+M_SEL) . . . . .	480
17.32	Loop Selector (Enable Tracking for primary loop only by CASIN_T) (M+M_SEL_T1) . . . . .	482
17.33	Loop Selector (Enable Tracking for primary loop by CASIN_T1/_T2) (M+M_SEL_T2) . . . . .	484
17.34	Loop Selector (Enable Tracking from secondary loop to primary loop) (M+M_SEL_T3_) . . . . .	487
17.35	Manual Output (M+M_MOUT) . . . . .	490
17.36	Program Setter (M+M_PGS) . . . . .	492
17.37	Multi-Point Program Setter (M+M_PGS2_) . . . . .	494
17.38	Manual Setter with Monitor (M+M_SWM_) . . . . .	498
17.39	Position Proportional Output (M+M_PVAL_T_) . . . . .	501
17.40	Heating and Cooling Output (M+M_HTCL_T_) . . . . .	510

---

**CHAPTER 18 STATUS TAG** **520**

18.1	Motor Irreversible (2 Input/2 Output) (M+M_NREV) . . . . .	520
18.2	Motor Reversible (2 Input/3 Output) (M+M_REV) . . . . .	524
18.3	ON/OFF Operation (2 Input/2 Output) (M+M_MVAL1) . . . . .	528
18.4	ON/OFF Operation (2 Input/3 Output) (M+M_MVAL2) . . . . .	532
18.5	Timer 1 (Timer stops when COMPLETE flag is ON) (M+M_TIMER1) . . . . .	536
18.6	Timer 2 (Timer continues when COMPLETE flag is ON) (M+M_TIMER2) . . . . .	538
18.7	Counter 1 (Counter stops when COMPLETE flag is ON) (M+M_COUNTER1) . . . . .	540
18.8	Counter 2 (Counter continues when COMPLETE flag is ON) (M+M_COUNTER2) . . . . .	543
18.9	Push Button Operation (5 Input/5 Output) (M+M_PB_) . . . . .	546

<b>CHAPTER 19 ALARM TAG</b>	<b>550</b>
19.1 Alarm (M+M_ALARM) .....	550
19.2 64 Points Alarm (M+M_ALARM_64PT_) .....	552

<b>CHAPTER 20 MESSAGE TAG</b>	<b>554</b>
20.1 Message (M+M_MESSAGE) .....	554
20.2 64 Points Message (M+M_MESSAGE_64PT_) .....	556

## **PART 6 PROCESS CONTROL INSTRUCTIONS**

<b>CHAPTER 21 PROCESS CONTROL INSTRUCTIONS</b>	<b>560</b>
<b>21.1 Overview</b> .....	<b>560</b>
Basic loop types .....	560
Process control instructions and data configuration .....	562
Data used by process control instructions .....	565
Loop tag memory assignments .....	568
How to execute process control instructions .....	571
Execution condition switching .....	573
Tracking .....	573
Precautions .....	576
<b>21.2 Lists of Process Control Instructions</b> .....	<b>578</b>
<b>21.3 I/O Control Instructions</b> .....	<b>583</b>
Analog input processing .....	583
Output processing 1 with mode switching .....	589
Output processing 2 with mode switching .....	595
Manual output .....	600
Time proportioning .....	603
Batch counter .....	609
Pulse integration .....	613
<b>21.4 Control Operation Instructions</b> .....	<b>618</b>
Basic PID control .....	618
Two-degree-of-freedom PID control .....	625
Position type PID control .....	632
Sample PI control .....	640
I-PD control .....	648
Blend PI control .....	655
Ratio calculation .....	662
Upper/lower limit alarm .....	667
Upper/lower limit alarm for power factor .....	673
Lead-lag compensation .....	681
Integral control .....	684
Derivative control .....	686
Dead time .....	689
High selector .....	692
Low selector .....	694
Middle value selector .....	696
Average value calculation .....	699
Upper/lower limiter .....	701
Variation rate limiter 1 .....	704

Variation rate limiter 2	707
Two-position (on/off) control	710
Three-position (on/off) control	716
Dead band	722
Program setter	724
Loop selector	729
Bumpless transfer	735
Analog memory	738
<b>21.5 Correction Operation Instructions</b>	<b>741</b>
Function generator	741
Inverse function generator	744
Standard filter	747
Integration	750
Temperature/pressure correction	752
Engineering value transformation	755
Engineering value inverse transformation	757
<b>21.6 Arithmetic Operation Instructions</b>	<b>759</b>
Addition	759
Subtraction	761
Multiplication	763
Division	765
Square root	767
Absolute value	769
<b>21.7 Comparison Operation Instructions</b>	<b>771</b>
Comparing data	771
<b>21.8 Auto Tuning</b>	<b>781</b>
Auto tuning instructions	784
<b>21.9 Lists of Loop Tag Memory</b>	<b>792</b>
PID control (SPID), two-degree-of-freedom PID control (S2PID), sample PI control (SSPI)	792
I-PD control (SIPD), blend PI control (SBPI)	794
Manual output (SMOUT), monitor (SMON)	796
Manual output with monitor (SMWM), PIDP control (SPIDP)	797
Two-position (on/off) control (SONF2), three-position (on/off) control (SONF3)	798
Batch counter (SBC)	799
Ratio control (SR)	800
<b>21.10 Processing Time</b>	<b>801</b>
Time added to instruction processing time	804
<b>21.11 Number of Basic Steps and Availability of Subset Processing</b>	<b>805</b>
<b>21.12 Program Examples</b>	<b>807</b>
<b>APPENDICES</b>	<b>812</b>
<hr/>	
<b>Appendix 1 Tag Data List</b>	<b>812</b>
PID	813
2PID	818
2PIDH	823
PIDP	829
SPI	833
IPD	837
BPI	841
R	845

ONF2	849
ONF3	852
MONI	855
SWM	857
MWM	860
BC	863
PSUM	865
SEL	866
MOUT	868
PGS	869
PGS2	872
PVAL	875
HTCL	879
NREV	882
REV	883
MVAL1	884
MVAL2	885
TIMER1	886
TIMER2	887
COUNT1	888
COUNT2	889
PB	890
ALM	891
ALM_64PT	892
MSG	899
MSG_64PT	900
Correspondence table of tag types and tag access FBs	907
Correspondence table of tag types and tag FBs	908
Correspondence table of tag types and alarms	909
Correspondence table of tag types and control modes	910
Correspondence table of tag types and I/O modes	911
<b>Appendix 2 Approximate Number of Steps</b>	<b>912</b>
List of number of steps	912
Increasing number of steps by the process control extension	915
<b>Appendix 3 Related Functions of Process</b>	<b>916</b>
Auto tuning	916
Various controls	923
I/O mode	928
Stop alarm processing in loop control	934
How to use the output open alarm	936
Program setter setting method	937
Tight shut/full open function	939
<b>Appendix 4 Program Example</b>	<b>940</b>
Control mode switching	940
Writing MV or SV with the host computer	942
Single solenoid	943
Double solenoid	944
<b>Appendix 5 Replacement of Other Format Projects (PX Developer-compatible Function Blocks)</b>	<b>945</b>
<b>Appendix 6 Program Example of Matching Operations with the Operations of PX Developer</b>	<b>947</b>
<b>Appendix 7 Replacement of Standard Function Blocks</b>	<b>949</b>

<b>Appendix 8 Version Upgrade</b> .....	<b>950</b>
Checking the version information .....	950
Version upgrade history .....	951

---

<b>INDEX</b>	<b>957</b>
--------------	------------

---

<b>INSTRUCTION INDEX</b>	<b>958</b>
--------------------------	------------

REVISIONS .....	962
WARRANTY .....	963
TRADEMARKS .....	964

# RELEVANT MANUALS

Manual name [manual number]	Description	Available form
MELSEC iQ-R Programming Manual (Process Control Function Blocks/Instructions) [SH-081749ENG] (this manual)	General process FBs, tag access FBs, tag FBs, and process control instructions designed for process control	e-Manual PDF
MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks) [SH-081266ENG]	Instructions for the CPU module and standard functions/function blocks	e-Manual PDF
MELSEC iQ-R Programming Manual (Module Dedicated Instructions) [SH-081976ENG]	Dedicated instructions for the intelligent function modules	e-Manual PDF
MELSEC iQ-R Programming Manual (Program Design) [SH-081265ENG]	Program specifications (ladder, ST, FBD/LD, and SFC programs)	e-Manual PDF
GX Works3 Operating Manual [SH-081215ENG]	System configuration, parameter settings, and online operations of GX Works3	e-Manual PDF
PX Developer Version 1 Operating Manual (Monitor Tool) [SH-080370E]	PX Developer monitoring tool and the monitoring and controlling of the DDC processing using the tag FB	Print book PDF

## Point

e-Manual refers to the Mitsubishi Electric FA electronic book manuals that can be browsed using a dedicated tool.

e-Manual has the following features:

- Required information can be cross-searched in multiple manuals.
- Other manuals can be accessed from the links in the manual.
- The hardware specifications of each part can be found from the product figures.
- Pages that users often browse can be bookmarked.
- Sample programs can be copied to an engineering tool.

# TERMS

Unless otherwise specified, this manual uses the following terms.

Term	Description
Assignment information data	The assignment information of the tag data assigned to the file register. The device information of the CPU module is stored.
Engineering tool	A tool used for setting up programmable controllers, programming, debugging, and maintenance
Faceplate	Gauge window on which an indicator such as a controller is displayed in image format. Monitoring the tag data and changing the current value can be performed.
FB property	The VAR_PUBLIC or VAR_PUBLIC_RETAIN class label among the local labels defined in FB. For the tag FB, the tag data member is included in the tag FB member. Setting the initial value for the FB property or changing the current value during a program execution changes the function block operation.
FBD/LD program for process control	An FBD/LD program with the extended function for the process control
Label	A label that represents a device in a given character string
Process control extension	Extending the functions of the engineering tool for the instrumentation engineering
Process control function block	In the engineering tool project of the Process CPU and the SIL2 Process CPU, the function block supplied by the manufacturer that can be used in the FBD/LD program with the process control extension enabled. The process control instruction corresponding to the Process CPU and the SIL2 Process CPU is executed in the function block.
Tag access FB	The process control function block that can be used only in the user-defined tag FB. Accessing the tag data of the user-defined tag FB executes the processing.
Tag data	The data where the process conditions and process status data associated with the tag FB are arranged. The engineering tool creates the tag data as a structure.
Tag FB	A function block with the process control function which works as a controller or an indicator. The tag data defined as the global label is used.
Tag type	Classification of the process control function which works as a controller or an indicator. The tag type determines the structure type of the tag data and the faceplate type.
User-defined tag FB	The function block that can customize the process control processing of the manufacturer-defined tag FB. The tag data can be referred from the FB program created by combining the function, function block, and process control function block.

# GENERIC TERMS AND ABBREVIATIONS

Unless otherwise specified, this manual uses the following generic terms and abbreviations.

Generic term and abbreviation	Description
SIL2 Process CPU	R08PSFCPU, R16PSFCPU, R32PSFCPU, R120PSFCPU. This module is used with a SIL2 function module as a pair, and performs both standard control and safety control. This module is also used with a redundant function module as a pair and configures a redundant system.
Process CPU	R08PCPU, R16PCPU, R32PCPU, R120PCPU



# MANUAL PAGE ORGANIZATION

In this manual, pages are organized and the symbols are used as shown below.

## How to read chapters 5 to 16 (process control function blocks)

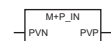
The following illustration is for explanation purpose only, and should not be referred to as an actual documentation.

### 9.1 Analog Input Processing (M+P\_IN)

➔ **1** **M+P\_IN**

This FB performs range check, input limiter, engineering value inverse conversion, and digital filter processing. The input limiter processing can be enabled or disabled on the "Options" window of the engineering tool.

➔ **2** **FBD/LD**



➔ **3** **Applicable tag type**

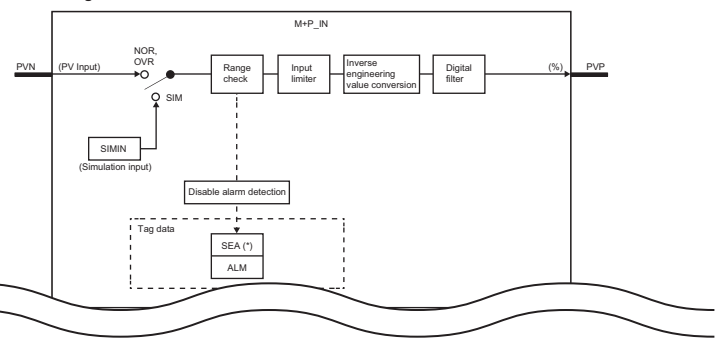
PID, 2PID, 2PIDH, PIDP, SPI, IPD, BPI, R, ONF2, ONF3, MONI, SWM, MWM, PVAL

➔ **4** **Control mode**

MAN	AUT	CAS <sup>*1</sup>	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

\*1 Transition to CASDR is possible.

➔ **5** **Block diagram**



➔ **6** **Setting data**

■ **Input/output variable**

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	NMIN to NMAX	Input variable	REAL
PVP	PV output	0 to 100[%]	Output variable	REAL

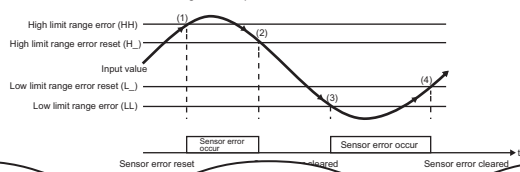
■ **Public variable (operation constant)**

Variable name	Description	Recommended range	Initial value	Set by	Data type
NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL

➔ **7** **Processing details**

■ **Range check**

This function block checks the range of an input value.



➔ **8** **Operation error**

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVN), block memory, operation constant (NMAX, NMIN, HH, H_, L_, LL) or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	High limit range error occurrence (HH) < high limit range error return (H_), low limit range error return (L_) < low limit range error occurrence (LL), or input high limit (NMAX) < input low limit (NMIN)

❶ Function block symbol

- A function block followed by parentheses indicates multiple function blocks. For example, "M+P\_HS(\_E)" includes two function blocks: M+P\_HS and M+P\_HS\_E.

❷ Description formats of the FBD/LD language

❸ Corresponding tag type (For tag access FB and tag FB)

❹ Corresponding control mode (For tag access FB and tag FB)

❺ Function block diagram

❻ Input/output variables and public variables

❼ Functions

❽ Error code and error details

# How to read chapter 17 (process control instructions)

The following illustration is for explanation purpose only, and should not be referred to as an actual documentation.

## Analog input processing

**1** → **S.IN**

This instruction performs following processing to the input data (PV): range check, input limiter, engineering value transformation, and digital filter.

**2** → **Ladder**

**ST**

ENO=S\_IN/EN.s1,s2,d1,d2;

**FBD/LD**

**3** → **Execution condition**

Instruction	Execution condition
S.IN	

### Setting data

**4** → **Description, range, data type**

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

**5** → **Applicable devices**

Operand	Bit		Word		Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□□□, J□□□, U3E□(H)G□	Z	LT, LST, LZ		K, H, E, \$			
(s1)	—	—	○	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—
(d2)	—	—	○	—	—	—	○	—	—	—	—

**6** → **Input data**

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value	-999999 to 999999	Single-precision real number	—	User
+1						

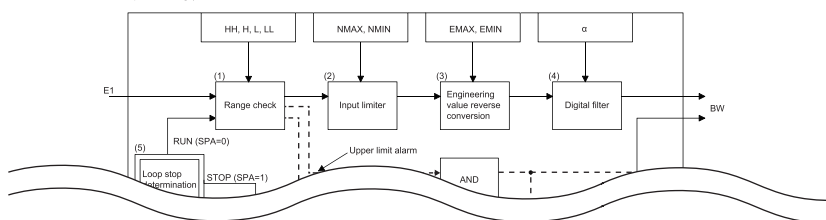
### Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	BW	Output value	-999999 to 999999 [%]	Single-precision real number	—	System
+1						
+2	BB	BB1 Alarm		16-bit unsigned binary	—	System
	BB2	Input upper limit alarm				
	BB3	Input lower limit alarm				

**7** → **Processing details**

This instruction performs engineering value transformation of the input value (E1) in the device specified by (s1), and stores the result in the device specified by (d1). The instruction also performs input value (E1) range check, input limiter, and digital filter processing.

The following is the processing block diagram of the S.IN instruction. (The numbers (1) to (5) in the diagram indicate the order of the processing.)



**8** → **Operation error**

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1), (d1), (s2), or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Upper limit range error occurrence (H#) < upper limit range error return (H), lower limit range error return (L) < lower limit range error occurrence (L), or input upper limit (NMAX) < input lower limit (NMIN)

① Instruction symbol

② Description formats of ladder diagram, structured text language, and FBD/LD

An instruction symbol should be described in the enclosed area of each ladder or FBD/LD program.

Execution condition is input to EN of each structured text or FBD/LD program. And, execution result should be described for ENO.

③ Execution condition (📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks))

④ Description of operands, setting ranges, and data types

⑤ Devices that can be used as operands

Operand	Bit		Word			Double word		Indirect specification	Constant			Others*5
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□*4	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□\G□, J□\□*4, U3E□\□(H)G□	Z	LT, LST, LC	LZ		K, H	E	S	
Applicable device*1	X, Y, M, L, SM, F, B, SB, FX*2, FY*2	J□\X J□\Y J□\B J□\SB	T*3, ST*3, C*3, D, W, SD, SW, FD*2, R, ZR, RD	U□\G□ U3E□\G□ U3E□\HG□ J□\W J□\SW	Z	LT*3 LST*3 LC*3	LZ	@□ @□.□	K, H	E		P, I, J, U, DX, DY, N, V

\*1 For details on each device, refer to the following.

📖 MELSEC iQ-R CPU Module User's Manual (Application)

\*2 FX and FY can be used for bit data only, and FD for word data only.

\*3 When T, ST, C, LT, LST, or LC is used for instructions, it can only be used as word data. It cannot be used as bit data.

\*4 This device can be used with a network module with a network number specified.

\*5 In the "Others" column, a device(s) that can be set for each instruction is shown.

⑥ Setting data. Some instructions require setting data that determine the operations of the instructions. When the loop memory, input data, block memory, operation constant, and local work memory need to be set by a user, set values according to the setting range. (📖 Page 565 Data used by process control instructions)

⑦ Processing details of the instruction. Unless otherwise specified, the following programs are regarded as interrupt programs.

- Interrupt program using the interrupt pointer (I)
- Fixed scan execution type program
- Event execution type program that is triggered by the interrupt pointer (I)

⑧ Error code and error details if the instruction has any possible operation error

- A device in which an error code is stored is provided in the error code column. When an error code is stored in SD0, an error flag (SM0) turns on. (The error status can be checked with the module label of the CPU module.)
- For the errors not provided here, refer to the following.

📖 MELSEC iQ-R CPU Module User's Manual (Application)

This part consists of the following chapters.

1 PROCESS CONTROL FUNCTION BLOCKS AND PROCESS CONTROL INSTRUCTIONS

2 PROCESS CONTROL FUNCTION BLOCK

3 PROCEDURE BEFORE USING PROCESS CONTROL FUNCTION BLOCK

# 1 PROCESS CONTROL FUNCTION BLOCKS AND PROCESS CONTROL INSTRUCTIONS

---

When a process control program is created, the process control instructions and the process control function blocks in which process control instructions are used.

A process control instruction and process control function block can be used with the Process CPU and SIL2 Process CPU (standard program only) for MELSEC iQ-R series.

## Point

When a process control program is created, using process control function blocks is recommended

Process control function blocks have features as follows.

- A process control program can be easily created by placing and connecting FB elements.
  - Since the initial value of the function block can be set in the "FB Property" window of the engineering tool, the program for the initial value setting is not required.
  - An operation constant can be input to a label indicating a tag name without being conscious of address of a device.
  - The operating status of a tag FB can be checked and controlled by accessing the tag data from the faceplate of an engineering tool.
- 

## Precautions

Process control function blocks are upgraded for improving or adding functionality at the time of version upgrade of GX Works3. For this reason, version of a function block used in a project which is created with an earlier GX Works3 version may not be the latest one.

When such a project is opened or read from the programmable controller, users may be prompted to upgrade the version of the process control function block. In this case, check the precautions and perform version upgrade. If a program for process control is changed without upgrading the version of the process control function block, unintended operations may result.

For the precautions of version upgrade and the version upgrade history, refer to the following.

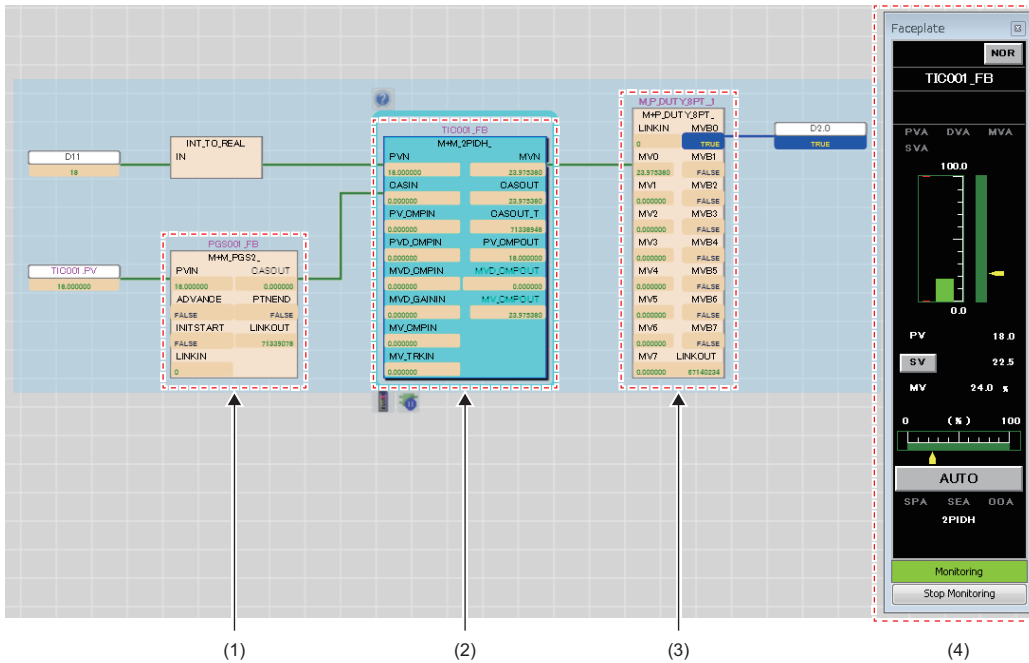
 Page 950 Version Upgrade

# 1.1 Process Control Function Block

A process control function block is a function block whose function is extended for process control. It contains structure data including process conditions and process status (tag data).

**Ex.**

Program using the tag FB (M+M\_2PIDH\_) which optimizes responsive performance for a setting value and control performance to a disturbance



- (1) Tag FB of multi-point program setter (M+M\_PGS2\_)
- (2) Tag FB of two-degree-of-freedom advanced PID control (M+M\_2PIDH\_)
- (3) General process FB of 8 points time proportional output (M+P\_DUTY\_8PT\_)
- (4) Faceplate of tag FB (M+M\_2PIDH\_)

For the overview of process control function blocks, refer to the following.

📖 Page 23 PROCESS CONTROL FUNCTION BLOCK

# 1.2 Process Control Instructions

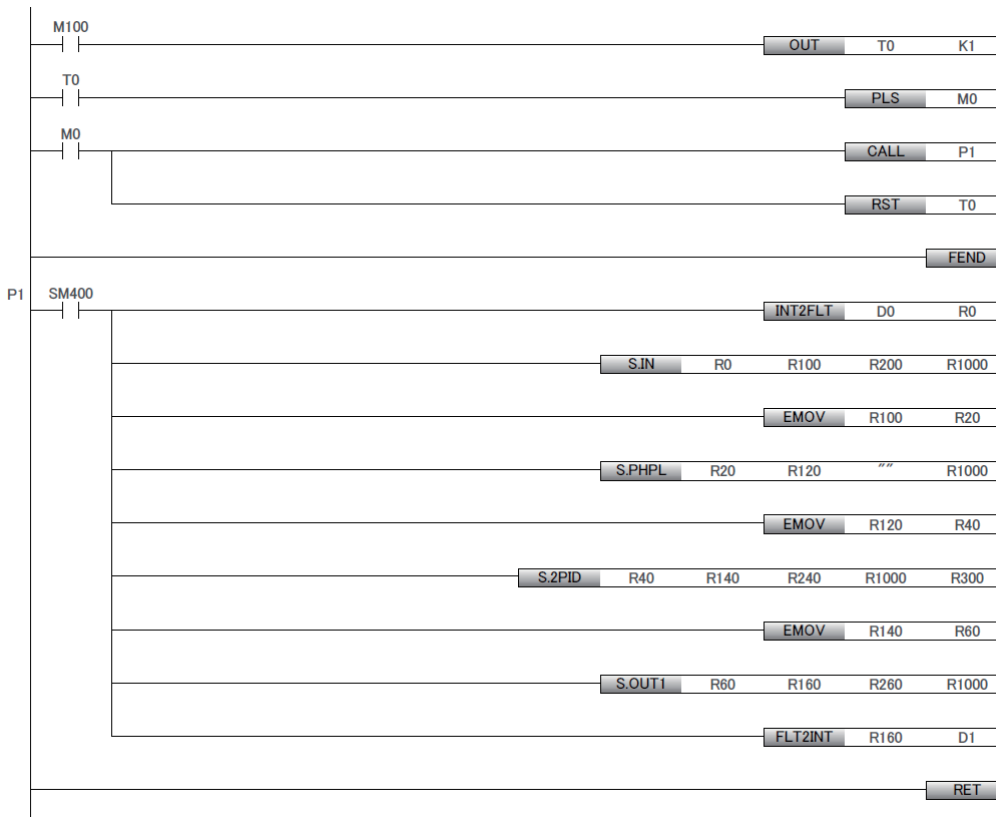
Combining these instructions that support loop control, such as two-degree-of-freedom PID control, sample PI, and auto tuning performs various types of process control.

**Ex.**

Program of two-degree-of-freedom PID control using the process control instruction

Settings of the loop tag memory and operation constants

⋮



Process control instructions have the following features:

- The process control instructions are available in the programs written in the ladder diagram, structured text, and FBD/LD.
- The process control instructions have the loop tag memories and each of them contains control information of each loop. Devices are assigned to each loop tag memory. Access to the assigned devices allows the settings of the initial values or checking of the execution status. (Use of the tag data is unavailable.)

For the process control instructions, refer to the following.

📖 Page 560 PROCESS CONTROL INSTRUCTIONS



# 2 PROCESS CONTROL FUNCTION BLOCK

## 2.1 Process Control Function Block Types

The process control function block has following types.

Item	Description
General process FB	Performs process control operation, such as correction operation and control operation.
Tag access FB	Performs processing by accessing tag data of a user-defined tag FB. It can be used only in the user-defined tag FB.
Tag FB	Performs process control operation as a controller or an indicator. It performs processing by accessing tag data defined as global labels. The execution status can be checked and controlled on the faceplate of the engineering tool.
User-defined tag FB	Combines tag access FBs, standard functions, or standard function blocks to extend process control processing of a tag FB. It performs processing by accessing tag data in the same way as a tag FB. The execution status can be checked and controlled on the faceplate of the engineering tool.

## 2.2 Program Supporting Process Control Function Blocks

A process control function block can be used in an FBD/LD program for process control.

A FBD/LD program for process control can be created when the process control extension is enabled on the "Properties" window of the program file.

The scan execution type or fixed scan execution type can be specified as an execution type of the FBD/LD program for process control. For the scan execution type, only one program file with process control extension enabled can be created.

The following table shows the availability of process control function blocks.

Item			Called function block					
			Local label			Global label		
			General process FB	Tag access FB	Tag FB, user-defined tag FB	General process FB	Tag access FB	Tag FB, user-defined tag FB
Calling source	Program	FBD/LD (process function extension enabled)	○	×	×	○	×	○ <sup>*2</sup>
		FBD/LD (process function extension disabled)	×	×	×	×	×	×
		Other than FBD/LD <sup>*3</sup>	×	×	×	×	×	×
	Function block	FBD/LD (user-defined tag FB)	○	○ <sup>*1</sup>	×	○	×	○ <sup>*2</sup>
		FBD/LD (other than user-defined tag FB) <sup>*4</sup>	○	×	×	○	×	○ <sup>*2</sup>
		Other than FBD/LD <sup>*3</sup>	×	×	×	×	×	×
Function		×	×	×	×	×	×	

\*1 The tag access FB can be used only in the user-defined tag FB.

\*2 The tag FB and the user-defined tag FB can be declared only as a global label. Declare them in the "Tag FB Setting" window of the engineering tool. (☞ Page 41 Declaring a tag FB (Tag FB Setting))

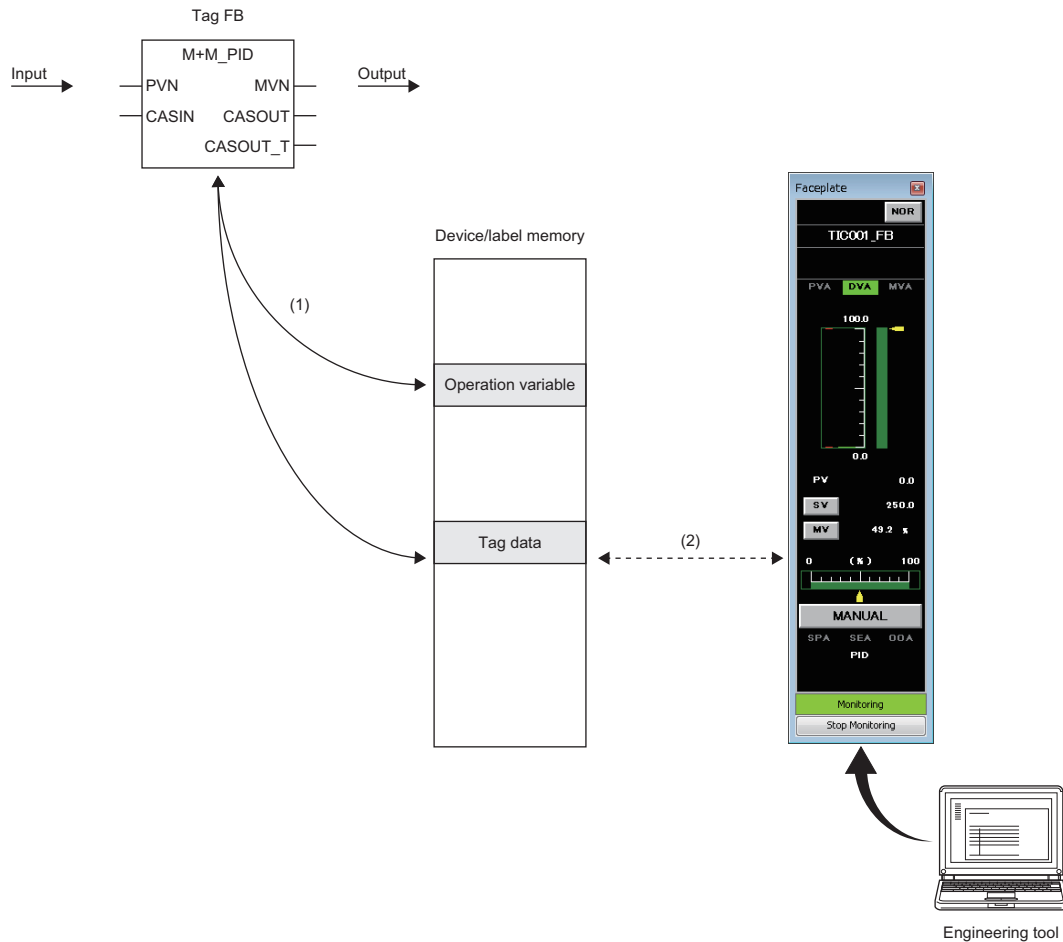
\*3 Actions and transitions in an SFC program are included.

\*4 Do not use a process control function block or a function block uses a user-defined tag FB in programs with the process control extension disabled. Doing so causes an error in programs.

## 2.3 Tag FB

A Tag FB and a user-defined tag FB perform process control operation as a controller or an indicator. They perform processing by accessing tag data defined as global labels.

The tag data defines data relevant to the instrumentation system as a structure. The execution status can be checked and controlled by accessing the tag data from the faceplate of the engineering tool.



- (1) The tag FB performs processing by accessing the tag data or operation constants (public variables of the FB).  
 (2) The execution status of the tag FB can be checked and controlled on the faceplate of the engineering tool.

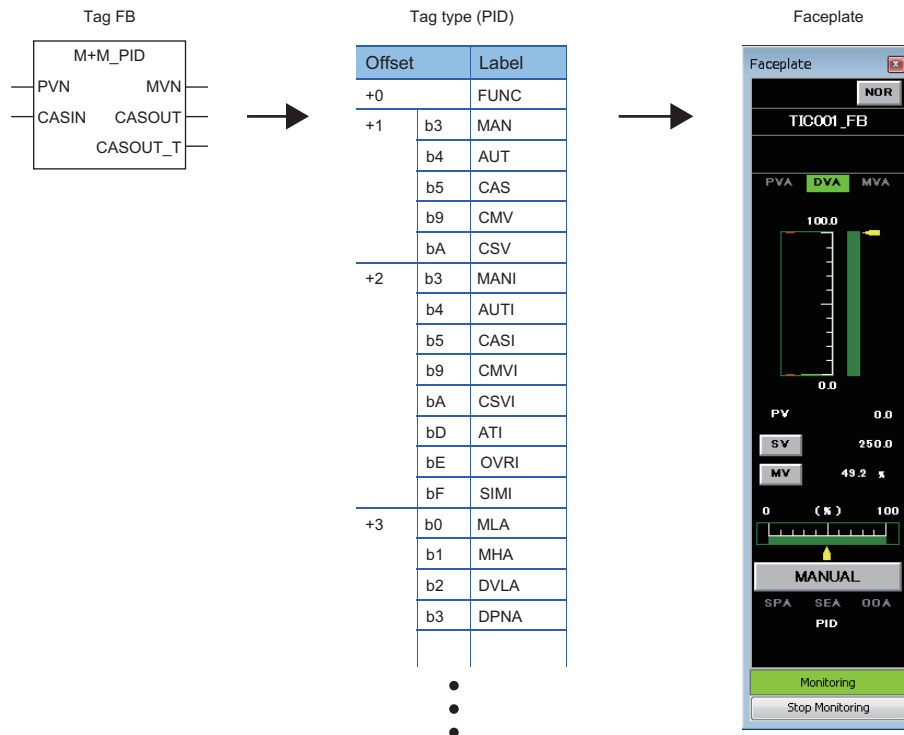
# Tag FB categories

The tag FB is classified into four categories according to the tag type. The following table lists the tag type classification.

Classification	Description
Loop tag	Used for loop control processing. This is equivalent to the loop tag used in process control instructions.
Status tag	Used for monitoring or control of the on/off states.
Alarm tag	Used to notify an alarm.
Message tag	Used to notify a message.

The tag data structure and faceplate type depend on the tag type of tag FB.

**Ex.**  
Tag type (PID) and faceplate for the tag FB (M+M\_PID)



For the tag type list, refer to the following.

📖 Page 29 User-defined Tag FB and Tag Access FB

## Initial values for tag data and operation constants (public variables)

The tag FB requires the initial value settings for tag data and operation constants (public variables).

The initial values for tag data and public variables of each tag FB part can be set on the "FB Property" window of the engineering tool. (🔗 Page 47 Setting Initial Values of FBs (FB Property))

## Control modes

The tag FB has six control modes.

Switching the control modes switches the tag FB control to the manual operation, automatic operation, or cascade operation.

The following table lists the control modes.

Symbol	Control mode	Description
MAN	MANUAL	Performs the manual operation. The MV setting value is output.
AUT	AUTO	Performs the automatic operation. The manipulated value is controlled on the basis of the SV setting value.
CAS	CASCADE	Performs the cascade operation. The output value (MV) of primary loop is controlled as the setting value (SV).
CMV	COMPUTER MV	Performs the manual operation with the host computer. The MV setting value of the host computer is output.
CSV	COMPUTER SV	Performs the automatic operation with the host computer. The MV value is controlled on the basis of the SV setting value of the host computer.
CASDR	CASCADE DIRECT	Directly outputs the output value of primary loop as that of secondary loop in the cascade connection.

The control mode can be switched with the faceplate of tag FB or M+P\_MCHG. (🔗 Page 378 M+P\_MCHG)

Available control modes depend on the tag type. (🔗 Page 910 Correspondence table of tag types and control modes)

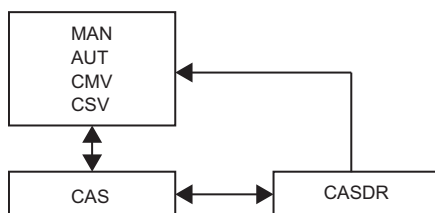
## Switching the control mode

### ■Control mode switching when the tag type is other than 2PIDH

There are no restrictions on the control mode switching.

### ■Control mode switching when the tag type is 2PIDH

The control mode can be switched to CASCADE DIRECT (CASDR) only from CASCADE (CAS). There are no restrictions on other control mode switching.



### Point

When the stop alarm (SPA) is TRUE, the control mode is automatically and forcibly switched to the MANUAL mode. (🔗 Stop alarm (SPA) overview)

## I/O modes

Inputs and outputs of the tag FB is connected or disconnected with the I/O module in accordance with the I/O mode setting. Also, the input processing and loop control operation of the tag FB can be stopped.

The tag FB has four I/O modes as follows.

Symbol	I/O mode	Description
NOR	NORMAL	Connects the signals from the I/O module (normal mode).
SIM	SIMULATION	Disconnects the signals from the I/O module and performs simulations. (☞ Page 928 Simulation function (SIMULATION mode))
OVR	OVERRIDE	Disconnects signals from the input module and enables the input of the process variable (PV) on the faceplate. This mode is used in case of the input sensor failure. (☞ Page 933 Override function (OVERRIDE mode))
TSTP	TAG STOP	Performs no processing related to the tag. The input processing and loop control operation are stopped. This mode is set for the tag which has been defined for future use or has stopped. All alarms related to the tag are reset, and no unnecessary alarm will occur. (☞ Page 933 Tag stop function (TAG STOP mode))

Available I/O modes depend on the tag type. For the correspondence between the tag types and I/O modes, refer to the following.

☞ Page 911 Correspondence table of tag types and I/O modes

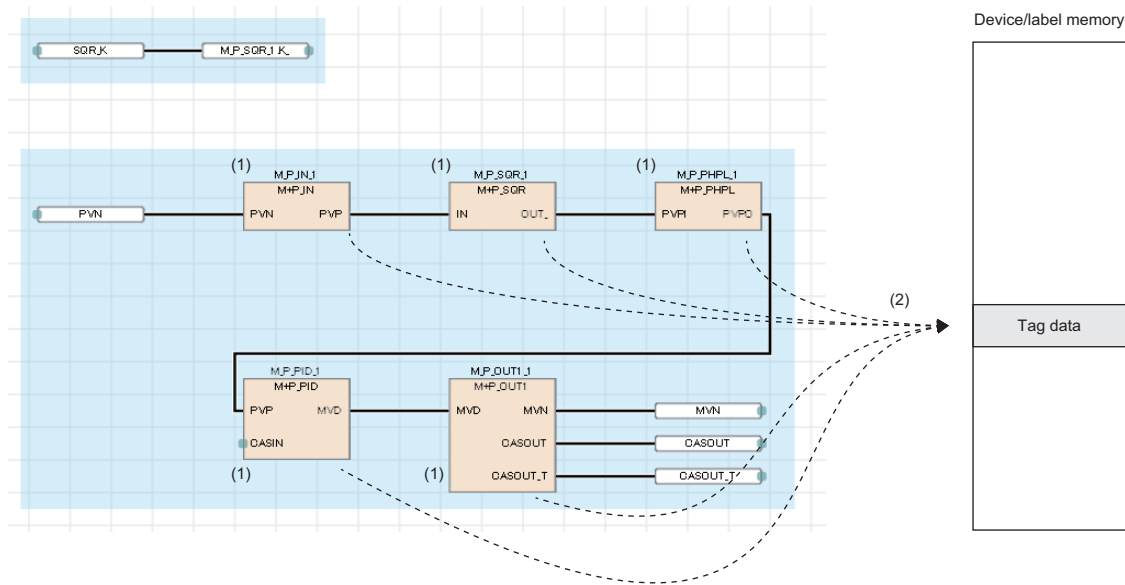
### Switching the I/O mode

The I/O mode can be switched when the control mode is MANUAL. Use the faceplate for switching. (☞ PX Developer Version 1 Operating Manual (Monitor Tool))

## 2.4 User-defined Tag FB and Tag Access FB

A tag FB with unique processing (user-defined tag FB) can be created with tag access FBs, standard functions, or standard function FBs.

The tag access FB performs processing by accessing tag data of the arranged user-defined tag FB.



(1) Tag access FB

(2) The tag access FB performs processing by accessing the tag data of user-defined tag FB.



The tag access FB can be used only in the user-defined tag FB.

# Tag type list

The following table lists the tag types, structure data types, structure labels for tag data reference, and application examples available in the user-defined tag FB.

Classification	Tag type	Name	Tag data structure type	Structure label for tag data reference	Application example in user-defined tag FB
Loop tag	PID	Basic PID control	M+TM_PID	_PID	_PID.MV
	2PID	Two-degree-of-freedom PID control	M+TM_2PID	_2PID	_2PID.MV
	2PIDH	Two-degree-of-freedom advanced PID control	M+TM_2PIDH	_2PIDH	_2PIDH.MV
	PIDP	Position type PID control	M+TM_PIDP	_PIDP	_PIDP.MV
	SPI	Sample PI control	M+TM_SPI	_SPI	_SPI.MV
	IPD	I-PD control	M+TM_IPD	_IPD	_IPD.MV
	BPI	Blend PI control	M+TM_BPI	_BPI	_BPI.MV
	R	Ratio control	M+TM_R	_R	_R.MV
	ONF2	Two-position (on/off) control	M+TM_ONF2	_ONF2	_ONF2.MV
	ONF3	Three-position (on/off) control	M+TM_ONF3	_ONF3	_ONF3.MV
	MONI	Monitor	M+TM_MONI	_MONI	_MONI.PV
	MWM	Manual output with monitor	M+TM_MWM	_MWM	_MWM.MV
	BC	Batch counter	M+TM_BC	_BC	_BC.PV
	PSUM	Pulse integration	M+TM_PSUM	_PSUM	_PSUM.PV
	SEL	Loop selector	M+TM_SEL	_SEL	_SEL.MV
	MOUT	Manual output	M+TM_MOUT	_MOUT	_MOUT.MV
	PGS	Program setter	M+TM_PGS	_PGS	_PGS.MV
	PGS2	Multi-point program setter	M+TM_PGS2	_PGS2	_PGS2.PV
	SWM	Manual setter with monitor	M+TM_SWM	_SWM	_SWM.MV
	PVAL	Position proportional output	M+TM_PVAL	_PVAL	_PVAL.PV
HTCL	Heating and cooling output	M+TM_HTCL	_HTCL	_HTCL.MV_HT	
Status tag	NREV	Motor irreversible control	M+TM_NREV	_NREV	_NREV.MAN
	REV	Motor reversible control	M+TM_REV	_REV	_REV.MAN
	MVAL1	On/off control 1 (without intermediate value)	M+TM_MVAL1	_MVAL1	_MVAL1.MAN
	MVAL2	On/off control 2 (with intermediate value)	M+TM_MVAL2	_MVAL2	_MVAL2.MAN
	TIMER1	Timer 1 (timer stops when COMPLETE flag is on)	M+TM_TIMER1	_TIMER1	_TIMER1.MAN
	TIMER2	Timer 2 (timer continues when COMPLETE flag is on)	M+TM_TIMER2	_TIMER2	_TIMER2.MAN
	COUNT1	Counter 1 (counter stops when COMPLETE flag is on)	M+TM_COUNT1	_COUNT1	_COUNT1.MAN
	COUNT2	Counter 2 (counter continues when COMPLETE flag is on)	M+TM_COUNT2	_COUNT2	_COUNT2.MAN
	PB	Push button operation	M+TM_PB	_PB	_PB.MAN
Alarm tag	ALM	Alarm	M+TM_ALM	_ALM	_ALM.ALM1
	ALM_64PT	64-points alarm	M+TM_ALM_64PT	_ALM_64PT	_ALM_64PT.ALM1
Message tag	MSG	Message	M+TM_MSG	_MSG	_MSG.MSG1
	MSG_64PT	64-points message	M+TM_MSG_64PT	_MSG_64PT	_MSG_64PT.MSG1



The available tag access FB parts depend on the tag type of user-defined tag FB. (Page 907 Correspondence table of tag types and tag access FBs)



# 2.5 Program Execution Control

An FBD/LD program for process control can be set a program execution cycle.

There are two methods for executing a program: timer execution and interrupt execution (fixed scan). Select either of the methods in accordance with the intended use of the program.

## Timer execution

An FBD/LD program for process control is executed in the execution cycle ( $\Delta T$ ) which is set to each program.

Among tag access FBs that structure of a loop tag FB, M+P\_IN, M+P\_PHPL, and M+P\_OUT1 used in I/O control are executed every execution cycle. Additionally, M+P\_PID and M+P\_2PID used in loop control operation are executed in the control cycle (CT) which is set to each tag. (🔗 Page 36 Control cycle (CT))

### Setting the execution cycle ( $\Delta T$ )

Select "High-Speed", "Normal-Speed", or "Low-Speed" for an execution cycle of each program block.

Set the execution cycle on the "Properties" window of the program block.

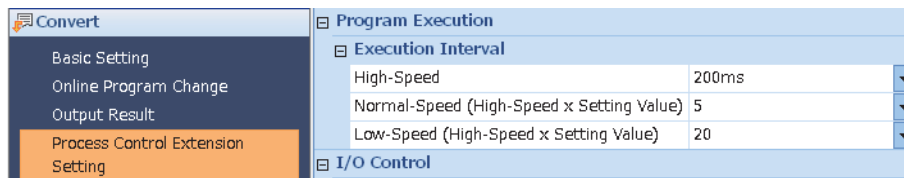


### Changing the execution cycle ( $\Delta T$ )

The following table lists the setting values and details of the execution cycle.

Execution cycle	Description	Default
High-Speed	"50ms", "100ms", or "200ms" can be selected for the high-speed execution cycle.	200ms
Normal-Speed	The normal-speed execution cycle is "High-speed execution cycle × Setting value". "2", "3", "4", or "5" can be selected for the normal-speed execution cycle.	1000ms (Setting value = 5)
Low-Speed	The low-speed execution cycle is "High-speed execution cycle × setting value". "5", "10", "20", "25", or "50" can be selected for the low-speed execution cycle.	4000ms (Setting value = 20)

The high-speed execution cycle and the setting values for normal/low-speed execution cycle can be set on the "Options" window of the engineering tool.



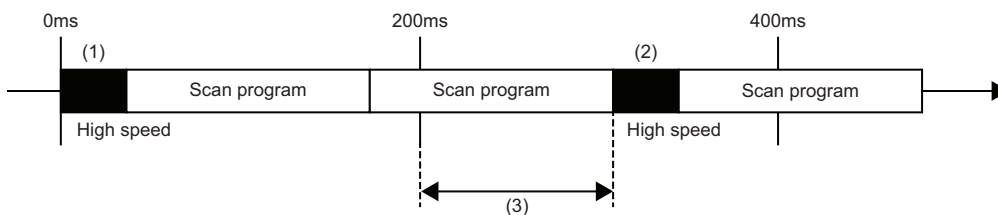
### ■Precautions

Set the scan time so that it is equal to or shorter than the high-speed execution cycle.

If the scan time exceeds the high-speed execution cycle, the fixed scan cycle cannot be maintained and an error of one scan time will occur at maximum in the timer execution program cycle. (🔗 Page 32 Phase and execution order)

**Ex.**

When the high-speed execution cycle is 200ms



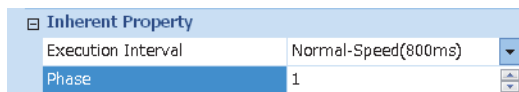
- (1) When the timer exceeds the specific cycle, the program block for process control is executed.
- (2) When the scan program is executed for the specific cycle or longer, the program block for process control is executed after the execution of the scan program ends.
- (3) The time to the end of the last scan program is an error. An error of one scan time can be occur at maximum.

## Phase and execution order

For the normal/low-speed execution cycle, set the phase.

With the phase setting, a program block can be executed at a timing shifted by a constant interval from the set execution cycle.

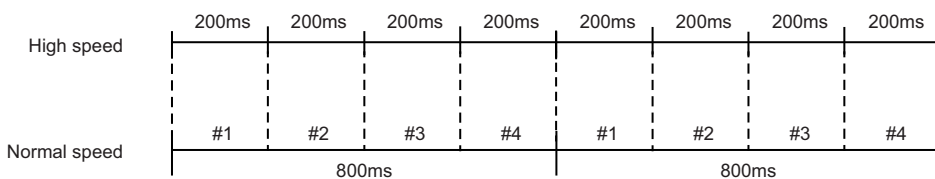
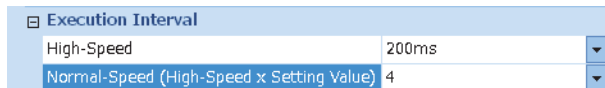
Set the phase on the "Properties" window of the program block.



**Ex.**

Phase for the normal-speed execution cycle of 800ms

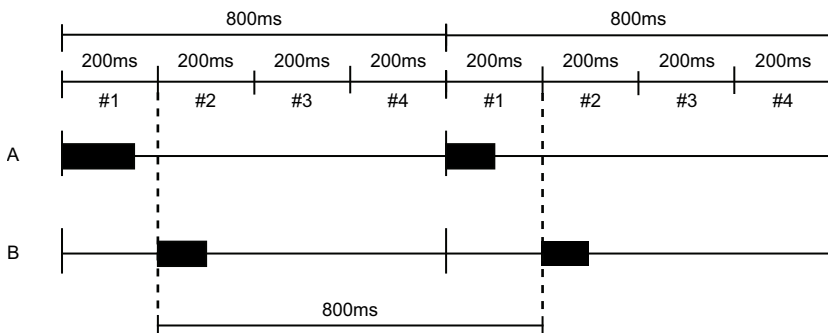
On the "Options" window of the engineering tool, set "200ms" to the high-speed execution cycle and "4" to the setting value of normal-speed execution cycle.



In this case, the execution cycle of 800ms is divided into four sections. The phases #1 to #4 can be set for the normal-speed program block.

**Ex.**

Fixed period execution of program block with phase

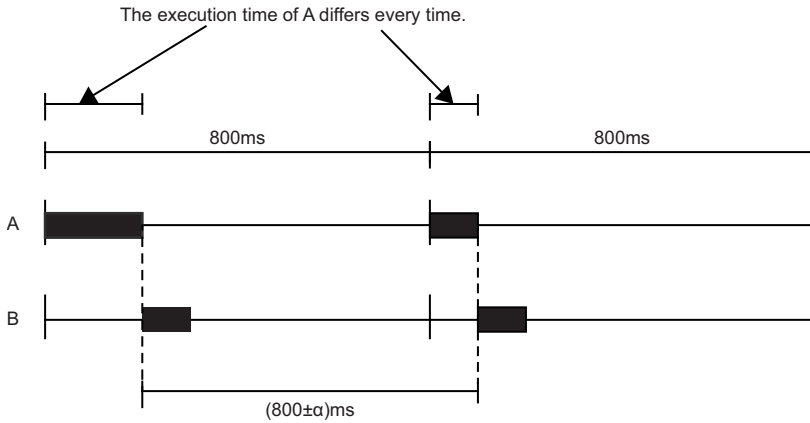


The execution cycle of 800ms is divided into four sections (200ms × 4). Set the program block A to be executed in the phase #1 and the program block B in the phase #2.

Consequently, when the execution time of the program block A is 200ms or shorter, the execution cycle of the program block B is always 800ms regardless of the execution time of the program block A.

Ex.

### Fixed period execution of program block without phase



The program blocks A and B have the same execution cycle of 800ms, and the program block A is first executed.

In this case, the program blocks are executed simultaneously, and the program block A is executed at an interval of 800ms and the program block B is executed after the execution of the program block A ends. Therefore, the execution time of program block A decides the execution cycle of the program block B, which will not be exactly 800ms. The on-time performance thus degrades.

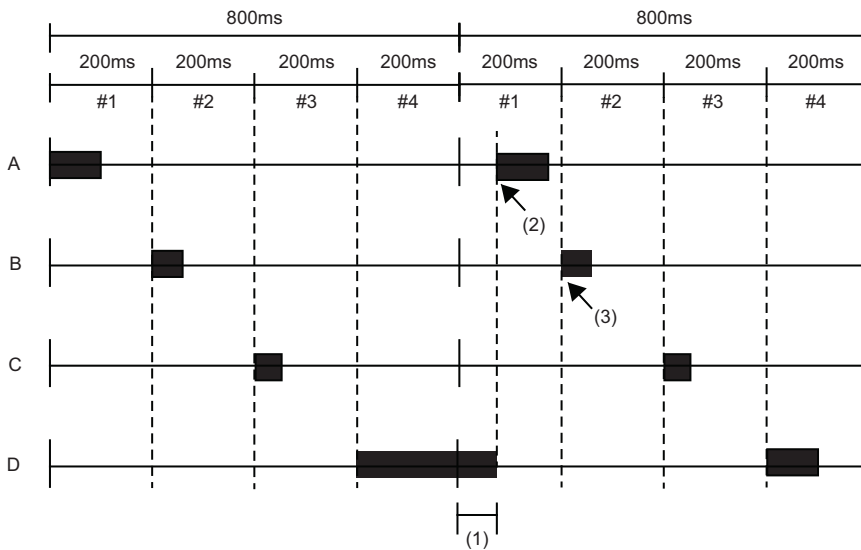
The undetermined factor ( $\pm\alpha$ ) in the execution cycle of the program block B results from the fluctuation of execution time of the program block A.

### ■ Program block with delay

If the program execution time exceeds the phase due to an interrupt execution, the next program block is not executed in the execution cycle, and the on-time performance may degrade.

**Ex.**

Fixed period execution of program block with delay



The program block A is executed in the phase #1, the program block B in the phase #2, the program block C in the phase #3, and the program block D in the phase #4.

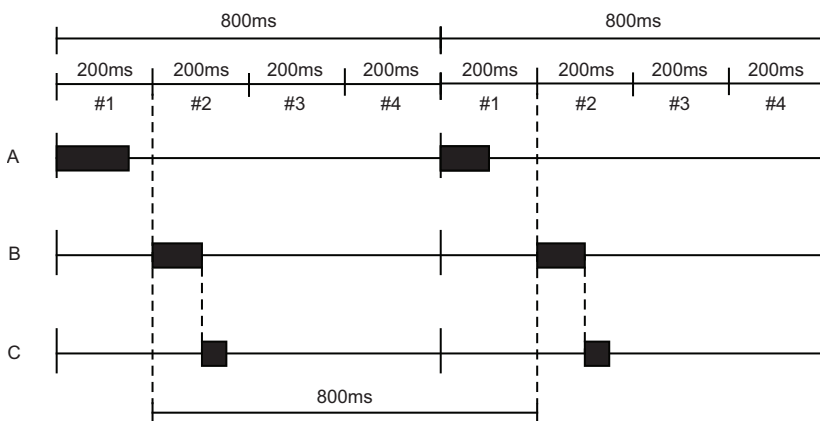
- (1) The execution time of the program block D is 200ms or longer due to an interrupt start program or other causes.
- (2) The program block A cannot be executed until the execution of the program block D ends. Therefore, the execution cycle of the program block A is not 800ms, and the on-time performance will degrade.
- (3) However, when the total sum of the exceeding time of the program block D and the execution time of the program block A is 200ms or shorter, the program block B is executed in the normal execution cycle.

### ■ Program block with the same execution cycle and phase

When multiple program blocks are set to be executed in the same phase, the program blocks are executed in the order set on the "Program File Setting" window of the engineering tool.

**Ex.**

Fixed period execution of program blocks with the same execution cycle and phase



Although the program blocks B and C have the same cycle and phase, the program block B is executed first in accordance with the execution order.

### ■Synchronizing phases in different execution cycles

The normal-speed program block is executed once when the high-speed program block is executed for "Normal-speed execution cycle ÷ High-speed execution cycle" times. The low-speed program block is executed once when the high-speed program block is executed for "Low-speed execution cycle ÷ High-speed execution cycle" times.

For example, when the execution cycle of high-speed program block is 100ms and that of normal-speed program block is 500ms, the normal-speed program block is executed once when the high-speed program block is executed five times.

Therefore, if the scan time exceeds 100ms, each program block cannot be executed in the setting cycle.

For example, when the execution cycle of normal-speed program block is 500ms and the scan time is 150ms, the high-speed program block is executed every 150ms. In this case, the normal-speed program block is executed once when the high-speed program block is executed five times. Consequently, the execution cycle of normal-speed program block is 750ms, and an error of 250ms (750ms - 500ms) will occur. Therefore, the scan time must be set to 100ms or shorter.

### ■Phase at operation change of the CPU module

A program is executed from the first phase when:

- CPU module is reset.
- CPU module is powered off and on.
- Operating status of the CPU module is switched from STOP to RUN.

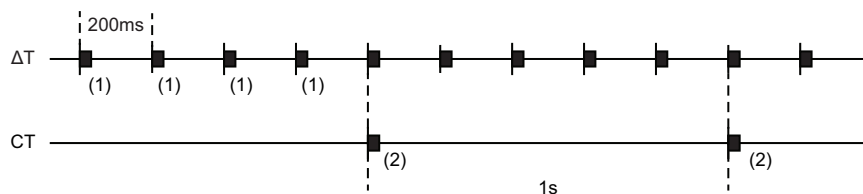
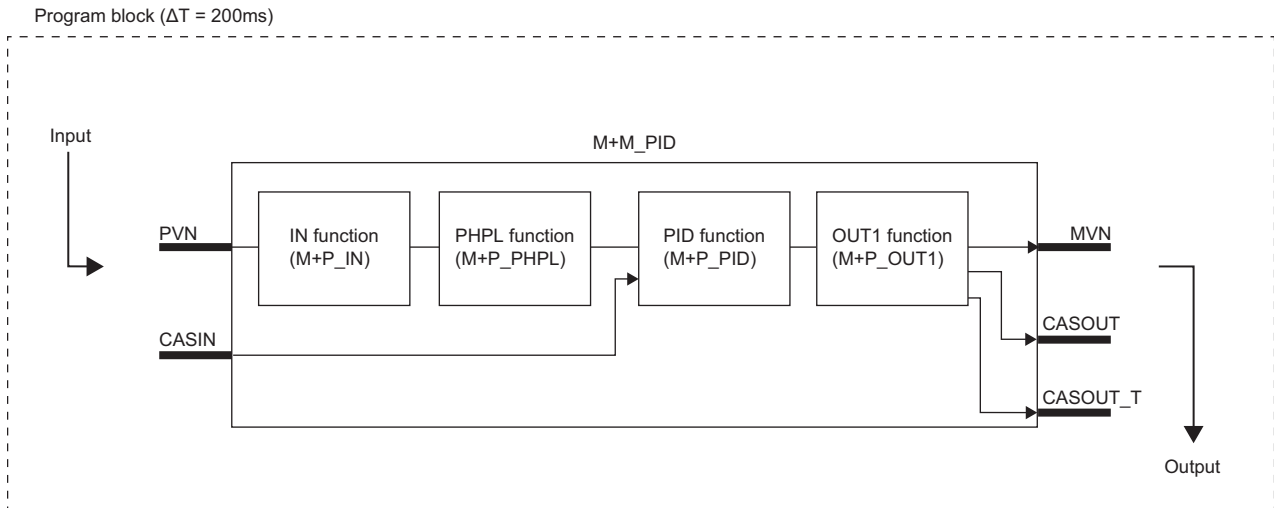
## Control cycle (CT)

A control cycle is the cycle in which the loop control operation is performed.

Set an integral multiple of the execution cycle as the control cycle for each tag. The execution cycles are counted, and the loop control operation is performed when the specified number of control cycles is reached.

**Ex.**

When the control cycle of M+M\_PID is 1s in the program block with the execution cycle of 200ms



(1) The IN function, PHPL function, and OUT1 function of M+M\_PID are executed every 200ms. The PID function is not executed and the last value is used for an output to OUT1.

(2) The IN function, PHPL function, PID function, and OUT1 function of M+M\_PID are executed.

If the control cycle is not the integral multiple of execution cycle, round off the number after the decimal point of control cycle  $(CT) \div$  execution cycle ( $\Delta T$ ) and multiply the execution cycle to calculate the control cycle.

For example, when the execution cycle ( $\Delta T$ ) is 1.0s and the control cycle (CT) is 2.5s, the calculation is  $2.5 \div 1.0 = 2.5 \rightarrow 3$ . Therefore, the control cycle is 3s.

## ■Setting the control cycle (CT)

Set the control cycle in the "FB Property" of the engineering tool.

Item	Name	Setting/Storage range		Unit	Initial value	Data type	Number of digits after the decimal point
		Low limit	High limit				
CT	Control cycle	0	9999	s	1.00	REAL	2

The following table lists the available tag types and tag access FBs performing operations every control cycle.

Tag type	Tag access FB
PID	M+P_PID(_T)
2PID	M+P_2PID(_T)
2PIDH	M+P_2PIDH(_T)_
PIDP	M+P_PIDP(_T), M+P_PIDP_EX(_T)_
IPD	M+P_IPD(_T)
BPI	M+P_BPI(_T)
R	M+P_R(_T)
ONF2	M+P_ONF2(_T)
ONF3	M+P_ONF3(_T)
SWM	M+P_MSET_
SEL	M+P_SEL_T3_

## Interrupt execution (fixed scan)

The interrupt execution interrupts the timer execution program and executes a program.

Set the program file as a fixed scan execution type program. The program file is executed at the fixed scan interval which has been set. (📖 MELSEC iQ-R CPU Module User's Manual (Application))

The fixed scan interval is recommended to set 10ms or more.

## When a safety program is used

The safety program is executed at the timing of the safety cycle time. The standard program (+ END processing) is executed in remaining time of the safety cycle time. Therefore, if the safety program + the safety input/output processing time is long, the execution cycle set for the FBD/LD program for process control may shift.

For the safety program and the safety cycle time, refer to the following.

(📖 MELSEC iQ-R CPU Module User's Manual (Application))

## When a fixed scan execution type program/an interrupt program is used

A fixed scan execution type program and an interrupt program may not always be executed at a specified timing.

- While a process control function block is being executed, interrupts are disabled by the DI instruction, and therefore a fixed scan execution type program or an interrupt program waits to run.
- Once a process control function block completes, interrupts are enabled by the EI instruction and a fixed scan execution type program/an interrupt program in a wait state runs.

This may affect high-speed pulse measurement with a CPU module and the result may be inaccurate. In this case, use the counter function of the pulse input module or the high-speed counter module.

When using these modules, refer to the manuals for each module.

For the fixed scan execution type program and interrupt program, refer to the following.

📖 MELSEC iQ-R CPU Module User's Manual (Application)

Check the processing time of a program being executed using the program list monitor function. Refer to the following.

📖 GX Works3 Operating Manual

# 3 PROCEDURE BEFORE USING PROCESS CONTROL FUNCTION BLOCK

This chapter describes how to create an FBD/LD program for process control using tag FBs.

## 3.1 Overview

### 1. Creating a project

Create a new project using the engineering tool. (📖 GX Works3 Operating Manual)

### 2. Setting CPU parameters

Set the following items in the CPU parameter.

- File Register Setting (👉 Page 39 File Register Setting)
- Index Register Setting (👉 Page 40 Index Register Setting)
- Label Initial Value Reflection Setting (👉 Page 40 Label Initial Value Reflection Setting)
- Tracking Setting (👉 Page 40 Tracking Setting)\*<sup>1</sup>

\*<sup>1</sup> The setting is required only when the operation mode of the Process CPU is set to "Redundant".

### 3. Declaring tag FBs (Tag FB Setting)

Declare tag FBs in the "Tag FB Setting" window. (👉 Page 41 Declaring a tag FB (Tag FB Setting))

### 4. Editing an FBD/LD program for process control

Drag and drop a tag FB from the element selection window and edit an FBD/LD program for process control. (👉 Page 42 Editing an FBD/LD Program for Process Control)

Create user-defined tag FBs as necessary. (👉 Page 45 Creating a user-defined tag FB)

### 5. Setting initial values of FBs (FB Property)

Set initial values of tag data and public variables (FB property) of tag FBs. (👉 Page 47 Setting Initial Values of FBs (FB Property))

### 6. Converting and writing

Perform "Rebuild All" (Reassignment) and write all the programs to the CPU module. (👉 Page 47 Converting and Writing)

### 7. Resetting the CPU module and executing programs

Reset the CPU module and switch the operating status to RUN. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

### 8. Checking FB operation (Faceplate)

Check the control status of a tag FB on a faceplate. (👉 Page 48 Checking FB Operation (Faceplate))

### 9. Reading the current value of FBs and reflecting the initial values (FB Property Management (Online))

The current value of the FB property can be read from the CPU module to be reflected to the initial value of the FB property with the "FB Property Management (Online)" function of the engineering tool. (📖 GX Works3 Operating Manual)



## 3.2 Setting CPU Parameters

Set the following items in the CPU parameter.

- File Register Setting
- Index Register Setting
- Label Initial Value Reflection Setting
- Tracking Setting\*<sup>1</sup>

\*<sup>1</sup> The setting is required only when the operation mode of the Process CPU is set to "Redundant".

### File Register Setting

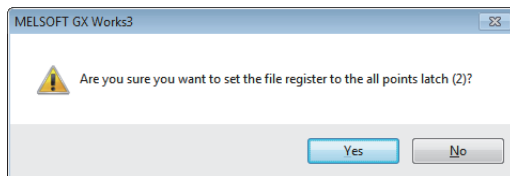
Set the file register.

 [CPU Parameter] ⇒ [File Setting] ⇒ [File Register Setting]

#### Operating procedure


File Register Setting	
<input type="checkbox"/> Use Or Not Setting	Use Common File Register in All Programs
Capacity	32 K Word
File Name	MAIN

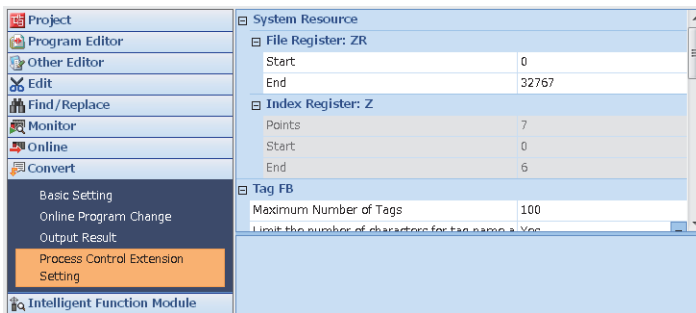
1. Select "Use Common File Register in All Programs" for "Use Or Not Setting".
2. Specify 32K words or more for "Capacity".
3. Enter any name in "File Name" and click the [Apply] button.
4. The window for setting the file register to the all points latch (2) is displayed. Click the [Yes] button.



#### Precautions

Set the value beyond the range specified in the option of the engineering tool as the file register capacity.

 [Tool] ⇒ [Options] ⇒ "Convert" ⇒ "Process Control Extension Setting" ⇒ "System Resource" ⇒ "File Register: ZR"



The range set in the option is used for assignment for the system area and tag data.

Therefore, do not use the setting range of the file register in the program.

However, it can be used for items of the public tag data.

When using the process control function block, do not use the QDRSET(P) instructions in the project. If the file name of file register is changed by the QDRSET(P) instructions, programs do not operate correctly.

# Index Register Setting

Check that seven points or more are set to the index register (Z). (The default setting is 20 points.)

 [CPU Parameter] ⇒ [Memory/Device Setting] ⇒ [Index Register Setting]




## Precautions

The range displayed in the option of the engineering tool (Z0 to Z6) is used for internal processing.

Therefore, do not use Z0 to Z6 of the index register in the program where the process control extension is enabled, and functions and function blocks used in that program.

# Label Initial Value Reflection Setting

Check that "Disable" is set to "Label Initial Value Reflection Setting at STOP to RUN" of "Label Initial Value Reflection Setting". (The default setting is "Disable".)

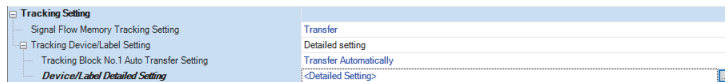
 [CPU Parameter] ⇒ [File Setting] ⇒ [Label Initial Value Reflection Setting]



# Tracking Setting

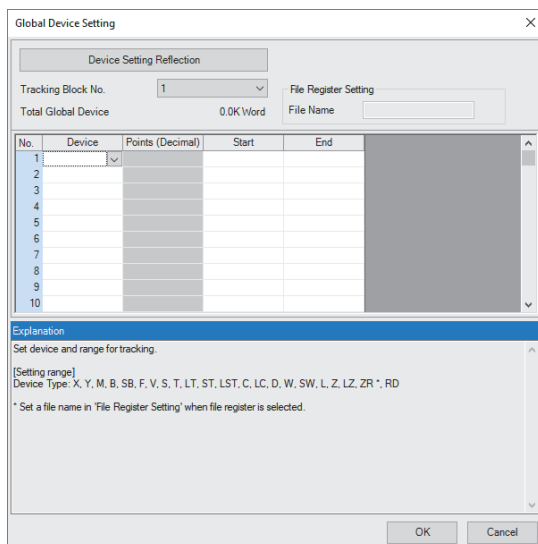
Set the parameters of the tracking transfer.

 [CPU Parameter] ⇒ [Redundant System Settings] ⇒ [Tracking Setting]



1. Select "Detailed setting" in "Tracking Device/Label Setting".
2. Open "Detailed Setting" of "Device/Label Detailed Setting".
3. Open "Global Device Setting".
4. The "Global Device Setting" window opens. Set devices other than the file registers to use in an FBD/LD program for process control. To reflect the devices and their areas that were set in "Device/Label Memory Area Setting", click the [Device Setting Reflection] button.

Tracking Block No.	Device Detailed Setting		Label Detailed Setting			Block Size Setting
	Global Device Setting	Local Device Setting	Global Label Setting	Local Label Setting	Module Label (Extension) Setting	
1	0.0K Word	Do Not Transfer	Do Not Transfer	Do Not Transfer	Do Not Transfer	0.0K Word
2	0.0K Word	Do Not Transfer	Do Not Transfer	Do Not Transfer	Do Not Transfer	0.0K Word
3	0.0K Word	Do Not Transfer	Do Not Transfer	Do Not Transfer	Do Not Transfer	0.0K Word



## Point

When the process control extension is enabled, "Tracking Device/Label Setting" is automatically set to "Detailed setting". Also, the file register area that was specified in "System Resource" of "Process Control Extension Setting" is entered in the tracking block No.64 and the corresponding tracking transfer trigger (bit 15 of SD1670) automatically turns on at every scan.

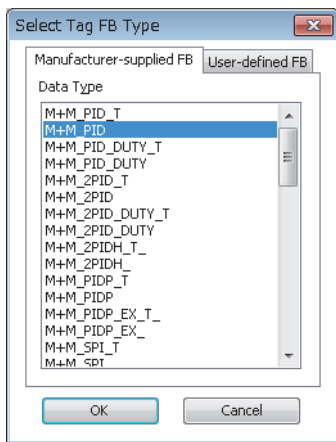
# 3.3 Declaring a tag FB (Tag FB Setting)

Declare a tag FB to be used in an FBD/LD program for process control.

## Operating procedure



No.	Tag Name	Tag FB Type	Tag Type
1			
2			
3			
4			



No.	Tag Name	Tag FB Type	Tag Type
1		M+M_PID	PID
2			
3			
4			

1. Click the "Tag FB Setting" button (1) on the toolbar to display the "Tag FB Setting" window.

When the "Tag FB Setting" button is not displayed on the toolbar, click [View] → [Toolbar] → [Process Control Extension].

2. Click (2) on the "Tag FB Setting" window to display the "Select Tag FB Type" window.

3. Select a data type of the tag FB to be used on the "Select Tag FB Type" window.

4. Click the [OK] button.

5. Enter "Tag Name" (3).

6. Click the [Apply] button.

After declaring a tag FB, FB instances and tag data are automatically registered to the global label. The related function blocks and structures of tag data are also automatically registered to the navigation window.

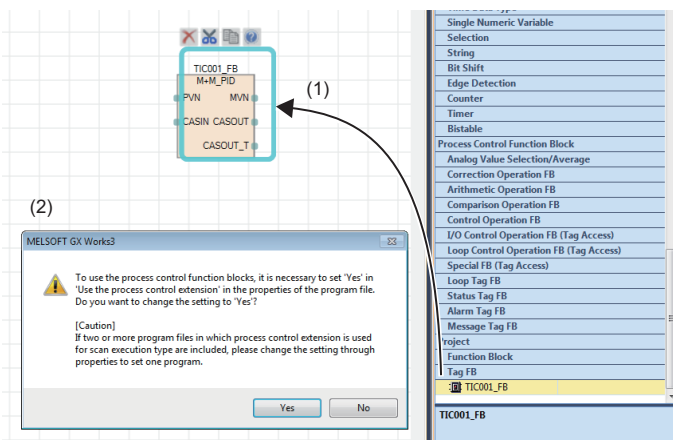
## Precautions

Before deleting a process control function block type, a tag data structure type, or a user-defined tag FB type, search the function block type and the structure type to be deleted with Find String and check that they are not used in the project. Deleting these data without checking may cause an error in the project.

# 3.4 Editing an FBD/LD Program for Process Control

Drag and drop a tag FB onto the FBD/LD editor and edit a program.

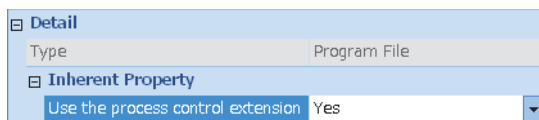
## Operating procedure



1. Select an instance of the declared tag FB on the element selection window and drag and drop it (1) to any place on the FBD/LD editor.
2. The setting dialog to use the process control function block (2) is displayed. Click the [Yes] button so that the process control function block can be used.
3. Arrange program elements on the input/output variables of the tag FB placed on the editor.

### Point

- Instances can be specified after dragging and dropping a tag FB from the navigation window or the element selection window onto the FBD/LD editor.
- For labels used in an FBD/LD program for process control, specify VAR\_RETAIN, VAR\_OUTPUT\_RETAIN, or VAR\_PUBLIC\_RETAIN label class to hold the data at power-off or reset of the CPU module.
- Setting for using process control function blocks (process control extension) can also be enabled on the "Properties" window of program files.



- The execution cycle and the phase can be set on the "Properties" window of program blocks. (Page 31 Timer execution)

## Specifying tag data and public variables

The following describes how to specify tag data and public variables in programs.

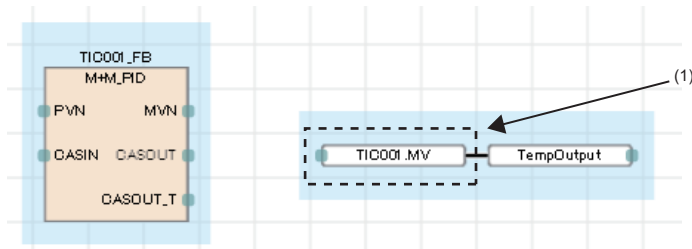
### ■ Specifying tag data

To use tag data of a tag FB in an FBD/LD program for process control, specify the tag data as "Tag name.Tag data name".

**Ex.**

When using the manipulated value (MV) of the tag data (TIC001)

No.	Tag Name	Tag FB Type	Tag Type	Structured Data Type	FB Instance Name	Label Name of Structured Data Type	Assigned Device
1	TIC001	M+M_PID	PID	M+TM_PID	TIC001_FB	TIC001	ZR3000



(1) Specify "TIC001.MV" using the tag name and tag data name of the tag data.

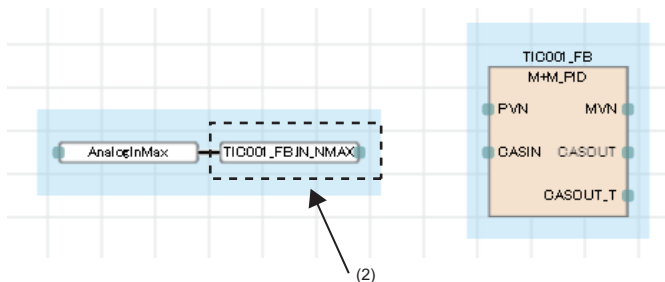
### ■ Specifying public variables

To use a public variable of a tag FB in an FBD/LD program for process control, specify the public variable as "FB instance name.Public variable name".

**Ex.**

When using the input high limit (IN\_NMAX) of the tag FB (TIC001\_FB)

No.	Tag Name	Tag FB Type	Tag Type	Structured Data Type	FB Instance Name	Label Name of Structured Data Type	Assigned Device
1	TIC001	M+M_PID	PID	M+TM_PID	TIC001_FB	TIC001	ZR3000



(2) Specify "TIC001\_FB.IN\_NMAX" using the FB instance name and member name of the tag FB.

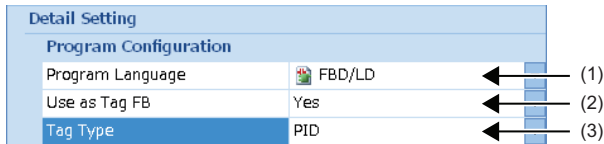
## Precautions

- Before placing a tag FB on the FBD/LD editor, declare the tag FB on the "Tag FB Setting" window.
- Do not use a process control function block in program files with the process control extension disabled. Doing so causes an error in programs.
- In a program and the function blocks/functions used in the program which are enabled the process control extension, the timer (T), retentive timer (ST), and timer FB (TIMER\_□\_M) do not correctly operate. Create a program using the long timer (LT), long retentive timer (LST), pulse timer (TP(\_E)), on delay timer (TON(\_E)), or off delay timer (TOF(\_E)).
- In a program and the function blocks/functions used in the program which are enabled the process control extension, set an execution cycle so that the count input signal turns on and off in a cycle longer than the execution cycle. If the count input signal of the counter (C), long counter (LC), or counter FB turns on and off in a cycle shorter than the execution cycle, the rising edge of the count input signal cannot be detected and an intended value may not be obtained.
- In a program and the function blocks/functions used in the program which are enabled the process control extension, a value obtained when a program is executed is stored in the special relay (SM).

# Creating a user-defined tag FB

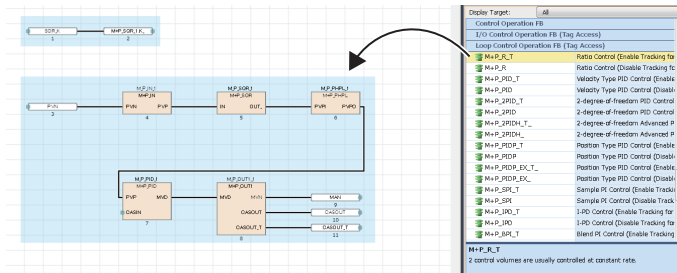
The following shows the procedure to create a user-defined tag FB.

## Operating procedure

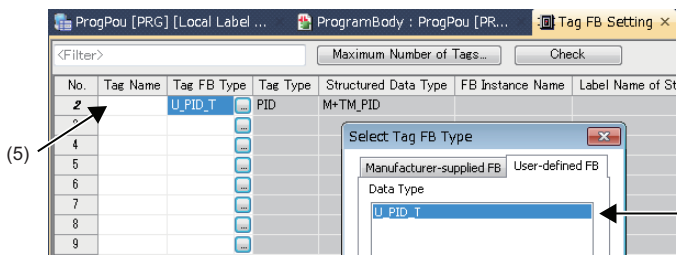


1. Create a new function block.
2. Set the items of "Program Configuration" as follows on the "New Data" window.
  - (1) Select "FBD/LD".
  - (2) Select "Yes".
  - (3) Select a tag type.

3

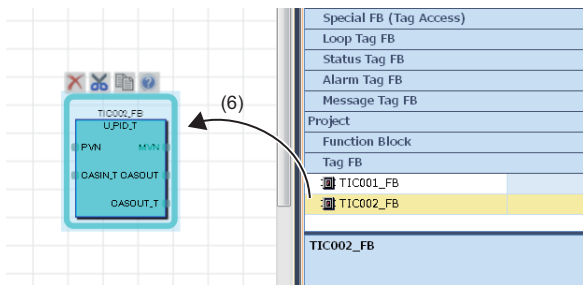


3. Create a program in a user-defined tag FB by combining tag access FBs, standard functions, or standard FBs.



4. Declare an instance of the user-defined tag FB on the "Tag FB Setting" window. Select the "User-defined FB" tab on the "Select Tag FB Type" window, and select the user-defined tag FB (4).

5. Enter a tag name (5) and click the [Apply] button.



6. In a FBD/LD program for process control, select an instance of the declared user-defined tag FB. Drag and drop it to the desired point on the FBD/LD editor (6).

### Point

A tag access FB available in a user-defined tag FB differs depends on the tag type. For the available tag access FBs, refer to the following.

☞ Page 907 Correspondence table of tag types and tag access FBs

## Specifying tag data of a user-defined tag FB

The following describes how to specify tag data of a user-defined tag FB to be used in a program.

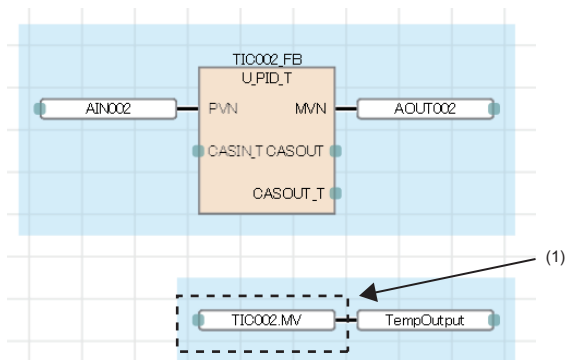
### ■When specifying the tag data from an FBD/LD program for process control

To use tag data of a user-defined tag FB in an FBD/LD program for process control, specify the tag data as "Tag name.Tag data name".

**Ex.**

When using the manipulated value (MV) of the tag data (TIC002)

No.	Tag Name	Tag FB Type	Tag Type	Structured Data Type	FB Instance Name	Label Name of Structured Data Type	Assigned Device
2	TIC002	U_PID_T	PID	M+TM_PID	TIC002_FB	TIC002	ZR3130



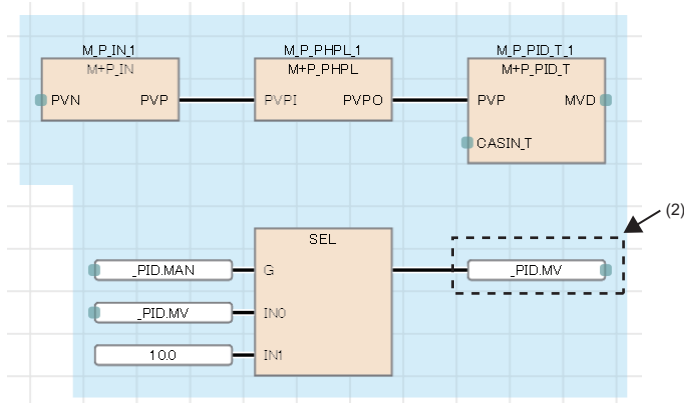
(1) Specify "TIC002.MV" using the tag name and tag data name.

### ■When specifying the tag data from a user-defined tag FB

To use tag data of a user-defined tag FB in the user-defined tag FB, specify the tag data as "Structure label for tag data reference.Tag data name".

**Ex.**

When using the manipulated value (MV) of the tag data in the user-defined tag FB "U\_PID\_T" (tag type: PID)



(2) Specify "\_PID.MV" using the structure label for tag data reference and tag data name.

For the structure labels for tag data reference, refer to the following.

☞ Page 30 Tag type list

## Precautions

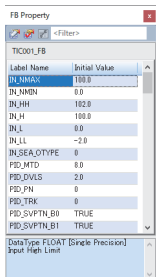
- Create a user-defined tag FB specifying the FB type to subroutine type.
- Do not use the EI instruction (enabling interrupt programs) in a user-defined tag FB or a function/function block called by the user-defined tag FB.
- Use a structure label for tag data reference in a user-defined tag FB only.




## 3.5 Setting Initial Values of FBs (FB Property)

Set initial values of tag data and public variables in the "FB Property" window.

### Operating procedure



1. Select a tag FB and display the "FB Property" window.  
 [View] ⇒ [Docking Window] ⇒ [FB Property]
2. Set initial values to the tag data or public variables.

The following initial values can be displayed or set on the "FB Property" window.

- VAR\_PUBLIC or VAR\_PUBLIC\_RETAIN class labels of the selected FB
- Members of tag data relevant to the selected FB

#### Point

To set initial values of public variables of the tag access FB arranged on a user-defined tag FB in "FB Property" of the engineering tool, refer to the following.

 GX Works3 Operating Manual

## 3.6 Converting and Writing


Perform "Rebuild All" (Reassignment)<sup>\*1</sup> and write the programs and their parameters to the CPU module.

- \*1 By performing "Rebuild All"/"Convert"/"Convert"+"Online Program Change", program blocks (M+PHEADER and M+PFOOTER) to be used by the system is added in a program file with the process control extension enabled. M+PHEADER and M+PFOOTER cannot be edited or deleted.

- For the converting of programs, refer to the following.

 GX Works3 Operating Manual

- For the writing of programs and parameters to the programmable controller, refer to the following.

 MELSEC iQ-R CPU Module User's Manual (Startup)

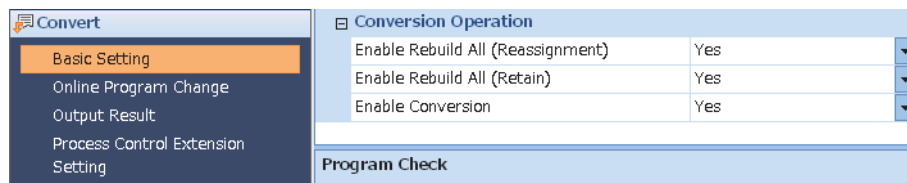
 GX Works3 Operating Manual

#### Point

To add or change program processing during system operation, perform "Convert" + "Online Program Change". Assignment and current values of labels that have been declared can be maintained.

Do not perform "Rebuild All" (retain) or "Convert".

"Rebuild All" (retain) and "Convert" can be disabled on the "Options" window of the engineering tool.

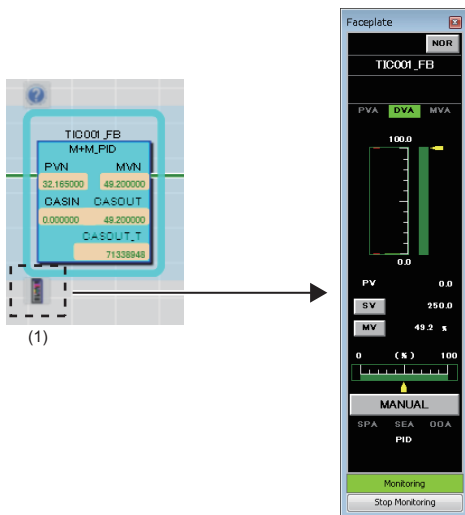


## 3.7 Checking FB Operation (Faceplate)

Set the operating status of the CPU module to RUN, and monitor and change the tag FB operation and current values.

### Operating procedure

1. Set the operating status of the CPU module to RUN to start monitoring.
2. Select a tag FB in an FBD/LD program for process control, and click (1).
3. Monitor the status of the tag FB on the faceplate, and add changes as necessary.



For the display configuration of the faceplate, refer to the following.

GX Works3 Operating Manual

## 3.8 Troubleshooting

When an error has occurred in the CPU module, check the error information by using the module diagnostics in the engineering tool. For details on the module diagnostics (CPU diagnostics), refer to the following.

GX Works3 Operating Manual

If the location where an error has occurred is a process control function block, check the detailed information in the module diagnostics window and explanations of error codes of each function block in this manual. Also, the detailed information of the error is stored in SD81 to SD111 (detailed information 1) and SD113 to SD143 (detailed information 2). For the information to be stored, refer to the following.

( MELSEC iQ-R CPU Module User's Manual (Application))

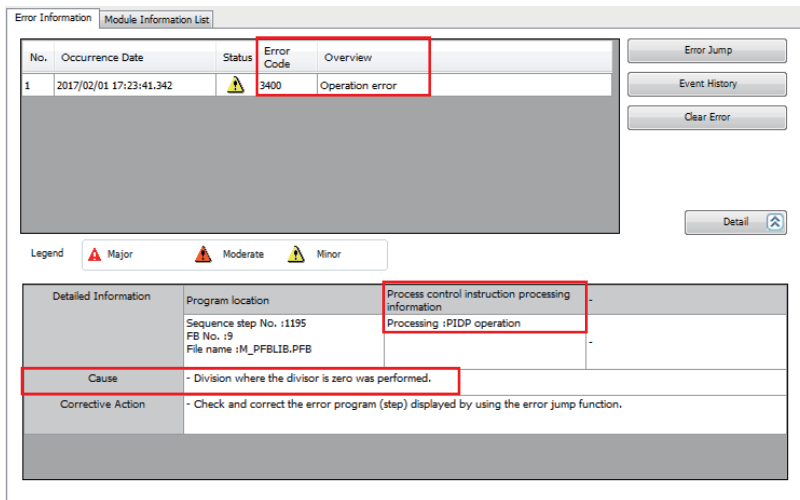
**Ex.**

Troubleshooting of when an error has occurred in a process control function block

1. Display the module diagnostics (CPU diagnostics) in the engineering tool.

[Diagnostics] ⇒ [Module Diagnostics (CPU Diagnostics)]

2. Check the error code and detailed information on the module diagnostics (CPU diagnostics) window.



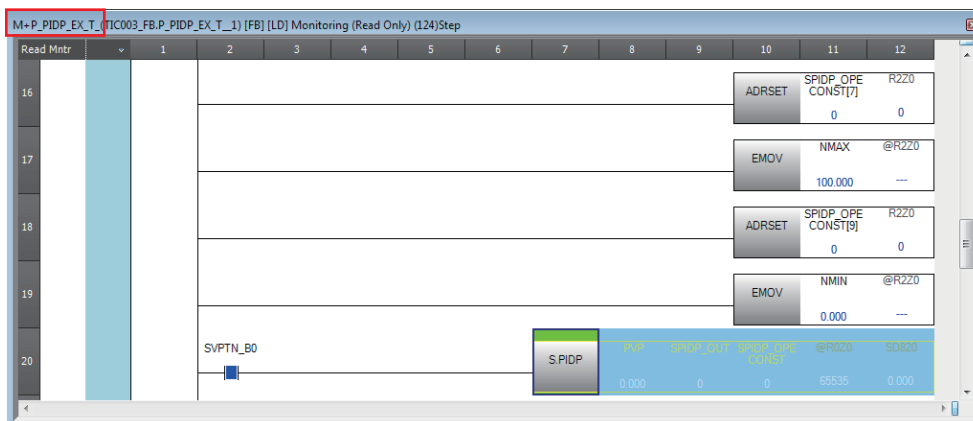
In the above example figure, the error code and detailed information are displayed as follows.

Error code: 3400

Processing: PIDP operation

Cause: Division where the divisor is zero was performed.

3. Click the [Error Jump] button to display the function block where the error has occurred. In the following example, the function block is M+P\_PIDP\_EX\_T\_.



4. Check the explanation of the corresponding function block in this manual.

Identify the location of the operation where caused the error and change current values of the input data, public variable, and tag data used in the operation.

In the case of error example above, check the current values, which may cause a division where the divisor is zero in a PIDP operation, of the following items in the function block of M+P\_PIDP\_EX\_T\_. Change the current value in the location where causes the error.

- Gain (PROPORTIONAL)
- Engineering value high limit (RH)
- Engineering value low limit (RL)
- Integral time (INTEGRAL)
- Derivative time (DERIVATIVE)
- Derivative gain (MTD)
- Control cycle (CT)



This part consists of the following chapters.

4 LISTS OF PROCESS CONTROL FUNCTION BLOCK


# 4 LISTS OF PROCESS CONTROL FUNCTION BLOCK

How to read the lists is shown below.

Item	Description
Function block symbol	A function block name is shown.
Processing details	An overview of the function blocks is explained.
Reference	Indicates the reference of detailed information.

## Point

- For instructions for the CPU module and standard functions/function blocks, refer to the following.

 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

- For dedicated instructions for the intelligent function modules, refer to the following.

 MELSEC iQ-R Programming Manual (Module Dedicated Instructions)

- For module FBs specific to each module, refer to the following.

 Function Block Reference for the module used

## 4.1 General Process FBs

### Analog Value Selection/Average

The following FBs output the maximum, minimum, middle, average, and absolute values of input values.

#### ■High Selector

Function block symbol	Processing details	Reference
M+P_HS(_E)	Outputs the maximum value of input values (array).	Page 72 M+P_HS(_E)

#### ■Low Selector

Function block symbol	Processing details	Reference
M+P_LS(_E)	Outputs the minimum value of input values (array).	Page 74 M+P_LS(_E)

#### ■Middle Value Selection

Function block symbol	Processing details	Reference
M+P_MID(_E)	Outputs the middle value of input values (array).	Page 76 M+P_MID(_E)

#### ■Average Value

Function block symbol	Processing details	Reference
M+P_AVE(_E)	Outputs the average value of input values (array).	Page 78 M+P_AVE(_E)

#### ■Absolute Value

Function block symbol	Processing details	Reference
M+P_ABS(_E)	Outputs the absolute value of input values.	Page 80 M+P_ABS(_E)

## Correction Operation

The following FBs perform operation processing such as broken line correction, standard filter, engineering value conversion, temperature/pressure correction, and summation.

### ■Function Generator

Function block symbol	Processing details	Reference
M+P_FG	Outputs (OUT_) values according to the function generator pattern that consists of SN break points with regard to the input (IN).	Page 82 M+P_FG

### ■Inverse Function Generator

Function block symbol	Processing details	Reference
M+P_IFG	Outputs (OUT_) values according to the function generator pattern that consists of SN break points with regard to the input (IN).	Page 85 M+P_IFG

### ■Standard Filter (moving average)

Function block symbol	Processing details	Reference
M+P_FLT	Outputs (OUT_) the average value of SN pieces of data of the input (IN) collected the at data collection interval ST.	Page 88 M+P_FLT

### ■Engineering Value Conversion

Function block symbol	Processing details	Reference
M+P_ENG	Converts a value (%) of the input (IN) in percentage into an engineering value such as temperature and pressure and outputs (OUT_) the result.	Page 90 M+P_ENG

### ■Engineering Value Inverse Conversion

Function block symbol	Processing details	Reference
M+P_IENG	Converts an input engineering value (IN) such as temperature and pressure into a value in percentage (%) and outputs (OUT_) the result.	Page 92 M+P_IENG

### ■Temperature/Pressure Correction

Function block symbol	Processing details	Reference
M+P_TPC	Executes temperature/pressure correction (or either of temperature correction or pressure correction) to the input (IN) of the differential pressure (%) and outputs (OUT_) the result.	Page 94 M+P_TPC

### ■Summation

Function block symbol	Processing details	Reference
M+P_SUM	When the integration start signal (START) is TRUE, this function block performs integration processing to the input (IN) and outputs (OUT_) the result.	Page 96 M+P_SUM

### ■Summation (Internal Integer Integration)

Function block symbol	Processing details	Reference
M+P_SUM2_	When the integration start signal (START) is TRUE, this function block performs integration processing to the input (IN) and outputs the result. Internal integration for the integral part is executed with signed 32-bit integers.	Page 98 M+P_SUM2_

### ■Range Conversion

Function block symbol	Processing details	Reference
M+P_RANGE_	Executes range conversion to the input (IN) and outputs (OUT_) the result.	Page 101 M+P_RANGE_

## Arithmetic Operation

The following FBs perform operation processing such as addition/subtraction, multiplication/division, and square root.

### ■ Addition (with coefficient)

Function block symbol	Processing details	Reference
M+P_ADD	Adds values with coefficients and bias to the inputs (IN1 to IN5) and outputs (OUT_) the result.	Page 103 M+P_ADD

### ■ Subtraction (with coefficient)

Function block symbol	Processing details	Reference
M+P_SUB	Subtracts values with coefficients and bias from the values of the inputs (IN1 to IN5) and outputs (OUT_) the result.	Page 105 M+P_SUB

### ■ Multiplication (with coefficient)

Function block symbol	Processing details	Reference
M+P_MUL	Multiplies the values of the inputs (IN1 to IN5) with coefficients and bias and outputs (OUT_) the result.	Page 107 M+P_MUL

### ■ Division (with coefficient)

Function block symbol	Processing details	Reference
M+P_DIV	Divides the values of the inputs (IN1, IN2) with coefficients and bias and outputs (OUT_) the result.	Page 109 M+P_DIV

### ■ Square Root (with coefficient)

Function block symbol	Processing details	Reference
M+P_SQR	Executes square root extraction with coefficients to the value of the input (IN) and outputs (OUT_) the result.	Page 111 M+P_SQR



## Comparison Operation

The following FBs perform comparison operation ( $\geq$ ,  $>$ ,  $=$ ,  $<$ ,  $\leq$ ).

### ■ Comparison ( $>$ ) with setting value

Function block symbol	Processing details	Reference
M+P_GT	Compares ( $>$ ) the input 1 (IN1) with the input 2 (IN2) using a setting value and hysteresis and outputs the result to the comparison output (B_). This function block always outputs values to the input 1 (IN1) from the output (OUT_).	Page 113 M+P_GT

### ■ Comparison ( $<$ ) with setting value

Function block symbol	Processing details	Reference
M+P_LT	Compares ( $<$ ) the input 1 (IN1) with the input 2 (IN2) using a setting value and hysteresis and outputs the result to the comparison output (B_). This function block always outputs values to the input 1 (IN1) from the output (OUT_).	Page 115 M+P_LT

### ■ Comparison ( $=$ ) with setting value

Function block symbol	Processing details	Reference
M+P_EQ	Compares ( $=$ ) the input 1 (IN1) with the input 2 (IN2) using a setting value and outputs the result to the comparison output (B_). This function block always outputs values to the input 1 (IN1) from the output (OUT_).	Page 117 M+P_EQ

### ■ Comparison ( $\geq$ ) with setting value

Function block symbol	Processing details	Reference
M+P_GE	Compares ( $\geq$ ) the input 1 (IN1) with the input 2 (IN2) using a setting value and hysteresis and outputs the result to the comparison output (B_). This function block always outputs values to the input 1 (IN1) from the output (OUT_).	Page 119 M+P_GE

### ■ Comparison ( $\leq$ ) with setting value

Function block symbol	Processing details	Reference
M+P_LE	Compares ( $\leq$ ) the input 1 (IN1) with the input 2 (IN2) using a setting value and hysteresis and outputs the result to the comparison output (B_). This function block always outputs values to the input 1 (IN1) from the output (OUT_).	Page 121 M+P_LE

## Control Operation

The following FBs control operation of lead-lag, integration, derivative, high/low limiter, variation rate limiter, dead band, bumpless transfer, and analog memory.

### ■Lead-Lag

Function block symbol	Processing details	Reference
M+P_LLAG	When the operation signal (INVLD) is FALSE, this function block performs lead-lag compensation to the input (IN) and outputs (OUT_) the result.	Page 123 M+P_LLAG

### ■Integration

Function block symbol	Processing details	Reference
M+P_I	When the operation signal (INVLD) is FALSE, this function block performs integral operation to the input (IN) and outputs (OUT_) the result.	Page 126 M+P_I

### ■Derivative

Function block symbol	Processing details	Reference
M+P_D	When the operation signal (INVLD) is FALSE, this function block performs derivative operation to the input (IN) and outputs (OUT_) the result.	Page 129 M+P_D

### ■Dead Time

Function block symbol	Processing details	Reference
M+P_DED	When the operation signal (INVLD) is FALSE, this function block outputs (OUT_) the result with a delay by the dead time to the input (IN).	Page 131 M+P_DED

### ■High/low Limiter

Function block symbol	Processing details	Reference
M+P_LIMT	Applies high/low limiters with hysteresis to the input (IN) and outputs (OUT_) the result.	Page 134 M+P_LIMT

### ■Variation Rate Limiter 1

Function block symbol	Processing details	Reference
M+P_VLMT1	Limits the variation speed to the input (IN) and outputs (OUT_) the result.	Page 137 M+P_VLMT1

### ■Variation Rate Limiter 2

Function block symbol	Processing details	Reference
M+P_VLMT2	Limits the output variation speed to the input (IN) and outputs (OUT_) the result.	Page 140 M+P_VLMT2

### ■Dead Band

Function block symbol	Processing details	Reference
M+P_DBND	Sets a dead band to the input (IN) and outputs (OUT_) the result.	Page 143 M+P_DBND

### ■Bump-less Transfer

Function block symbol	Processing details	Reference
M+P_BUMP	Changes the output (OUT_) from the output control value CTRLV to the output setting value SETV smoothly when the mode (MODE) is changed from FALSE (MANUAL) to TRUE (AUTO).	Page 145 M+P_BUMP

### ■Analog Memory

Function block symbol	Processing details	Reference
M+P_AMR	Increases or decreases the output (OUT_) at a fixed rate.	Page 147 M+P_AMR

### ■8 Points Time Proportional Output

Function block symbol	Processing details	Reference
M+P_DUTY_8PT_	Performs output ON time conversion for input values and outputs the result in bits. This function block also adjusts the phase of output cycles automatically to suppress overlapping of output (for example, peak current).	Page 149 M+P_DUTY_8PT_

### ■ Bistable Switch (Set Dominant)

Function block symbol	Processing details	Reference
M+SR	Bistable Switch (Set Dominant) (When S1_ and R_ are TRUE simultaneously, Q1 is TRUE.)	Page 155 M+SR

### ■ Bistable Switch (Reset Dominant)

Function block symbol	Processing details	Reference
M+RS	Bistable Switch (Reset Dominant) (When S_ and R1_ are TRUE simultaneously, Q1 is FALSE.)	Page 157 M+RS

### ■ Latch FB (BOOL Type)

Function block symbol	Processing details	Reference
M+LATCH_BOOL	When E_=TRUE, outputs IN to OUT_, and when E_=FALSE, for OUT_, latches an output.	Page 159 M+LATCH_BOOL

### ■ Latch FB (REAL Type)

Function block symbol	Processing details	Reference
M+LATCH_REAL	When E_=TRUE, outputs IN to OUT_, and when E_=FALSE, for OUT_, latches an output.	Page 160 M+LATCH_REAL

### ■ Latch FB (WORD Type)

Function block symbol	Processing details	Reference
M+LATCH_WORD	When E_=TRUE, outputs IN to OUT_, and when E_=FALSE, for OUT_, latches an output.	Page 161 M+LATCH_WORD

### ■ Latch FB (DWORD Type)

Function block symbol	Processing details	Reference
M+LATCH_DWORD	When E_=TRUE, outputs IN to OUT_, and when E_=FALSE, for OUT_, latches an output.	Page 162 M+LATCH_DWORD

### ■ Rising Edge Detector

Function block symbol	Processing details	Reference
M+R_TRIG	When the rising edge (CLK) is detected, Q is TRUE while this function block is executed once.	Page 163 M+R_TRIG

### ■ Falling Edge Detector

Function block symbol	Processing details	Reference
M+F_TRIG	When the falling edge (CLK) is detected, Q is TRUE while this function block is executed once.	Page 164 M+F_TRIG

### ■ Edge Detector

Function block symbol	Processing details	Reference
M+EDGE_CHECK	When the rising edge (CLK1) or the falling edge (CLK2) is detected, OUT_ is TRUE while this function block is executed once.	Page 165 M+EDGE_CHECK

### ■ Up Counter

Function block symbol	Processing details	Reference
M+CTU	Counts up by the external input.	Page 166 M+CTU

### ■ Down Counter

Function block symbol	Processing details	Reference
M+CTD	Counts down by the external input.	Page 168 M+CTD

### ■ Up/Down Counter

Function block symbol	Processing details	Reference
M+CTUD	Counts up/down by the external input.	Page 169 M+CTUD

### ■ Pulse Timer (High-speed Timer Type)

Function block symbol	Processing details	Reference
M+TP_HIGH	Pulse timer using a high-speed timer	Page 171 M+TP_HIGH

### ■ Pulse Timer (Low-speed Timer Type)

Function block symbol	Processing details	Reference
M+TP_LOW	Pulse timer using a low-speed timer	Page 173 M+TP_LOW

### ■ ON Delay Timer (High-speed Timer Type)

Function block symbol	Processing details	Reference
M+TON_HIGH	ON delay timer using a high-speed timer	Page 175 M+TON_HIGH

### ■ ON Delay Timer (Low-speed Timer Type)

Function block symbol	Processing details	Reference
M+TON_LOW	ON delay timer using a low-speed timer	Page 177 M+TON_LOW

### ■ OFF Delay Timer (High-speed Timer Type)

Function block symbol	Processing details	Reference
M+TOF_HIGH	OFF delay timer using a high-speed timer	Page 179 M+TOF_HIGH

### ■ OFF Delay Timer (Low-speed Timer Type)

Function block symbol	Processing details	Reference
M+TOF_LOW	OFF delay timer using a low-speed timer	Page 181 M+TOF_LOW

## 4.2 Tag Access FBs

### I/O Control Operation

The following FBs perform I/O processing including analog input/output, pulse integration, and batch counter.

#### ■Analog Input Processing

Function block symbol	Processing details	Reference
M+P_IN	Performs range check, input limiter, engineering value inverse conversion, and digital filter processing. The input limiter processing can be enabled or disabled on the "Options" window of the engineering tool.	Page 184 M+P_IN

#### ■Output Processing-1 with Mode Switching (With Input Addition)

Function block symbol	Processing details	Reference
M+P_OUT1	Performs input addition, variation rate & high/low limiter, reset windup, and output conversion processing to an input value ( $\Delta MV$ ) and outputs a manipulated value. (with integral and anti-reset windup processing)	Page 189 M+P_OUT1

#### ■Output Processing-2 with Mode Switching (Without Input Addition)

Function block symbol	Processing details	Reference
M+P_OUT2	Performs variation rate & high/low limiter processing and output conversion processing to an input value (MV), and outputs a manipulated value. (without integral and anti-reset windup processing)	Page 194 M+P_OUT2

#### ■Output Processing-3 with Mode Switching

Function block symbol	Processing details	Reference
M+P_OUT3_	Performs input addition, MV compensation, preset MV, MV hold, MV tracking, variation rate & high/low limiter, reset windup, tight shut/full open, MV reverse, and output conversion processing to an input value ( $\Delta MV$ ) and outputs a manipulated value.	Page 198 M+P_OUT3_

#### ■Manual Output

Function block symbol	Processing details	Reference
M+P_MOUT	Reads a manipulated value (MV) of tag data, performs output conversion processing, and outputs a manipulated value.	Page 206 M+P_MOUT

#### ■Time Proportioning Output

Function block symbol	Processing details	Reference
M+P_DUTY	Performs input addition, variation rate & high/low limiter, reset windup, output ON time conversion, and output conversion processing to an input value ( $\Delta MV$ ) and outputs the value as bits.	Page 208 M+P_DUTY

#### ■Pulse Integrator

Function block symbol	Processing details	Reference
M+P_PSUM	When the integration start signal (RUN) is TRUE, this function block performs input value increment operation, integrated value calculation, and output conversion processing to a count value (CIN) and outputs the result.	Page 213 M+P_PSUM

#### ■Batch Counter

Function block symbol	Processing details	Reference
M+P_BC	Compares the input (CIN) with the estimated value 1 and estimated value 2, and outputs a completion signal when the input reaches an estimated value. At this time, this function block performs the high limit check, variation rate check, and output conversion processing to the input (CIN).	Page 216 M+P_BC

#### ■Manual Setter

Function block symbol	Processing details	Reference
M+P_MSET_	Performs SV variation rate & high/low limiter processing, sets the result as the set value (current) (SVC) of tag data, and outputs the value (SVC).	Page 219 M+P_MSET_

## Loop Control Operation

The following FBs perform loop control processing including ratio control, various PID controls, two-position (on/off) control, three-position (on/off) control, program setter, and loop selector.

### ■Ratio Control (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+P_R_T	Controls two sets of control amount at a constant ratio and outputs the result ( $\Delta MV$ ). The primary loop can be tracked.	Page 223 M+P_R_T

### ■Ratio Control (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+P_R	Controls two sets of control amount at a constant ratio and outputs the result ( $\Delta MV$ ).	Page 227 M+P_R

### ■Velocity Type PID Control (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+P_PID_T	Performs PID operations using process variable differential, inexact differential, and velocity type, and outputs the result ( $\Delta MV$ ). The primary loop can be tracked.	Page 230 M+P_PID_T

### ■Velocity Type PID Control (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+P_PID	Performs PID operations using process variable differential, inexact differential, and velocity type, and outputs the result ( $\Delta MV$ ).	Page 236 M+P_PID

### ■2-degree-of-freedom PID Control (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+P_2PID_T	Optimizes the responsive performance (tracking performance) in a setting value change and control performance to a disturbance, and outputs the result ( $\Delta MV$ ). The primary loop can be tracked.	Page 242 M+P_2PID_T

### ■2-degree-of-freedom PID Control (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+P_2PID	Optimizes the responsive performance (tracking performance) in a setting value change and control performance to a disturbance, and outputs the result ( $\Delta MV$ ).	Page 248 M+P_2PID

### ■2-degree-of-freedom Advanced PID Control (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+P_2PIDH_T_	Optimizes the responsive performance (tracking performance) in a setting value change and control performance to a disturbance, and outputs the result ( $\Delta MV$ ). This function block performs two-degree-of-freedom PID operation, PV tracking, integration stop, derivative stop, and SV variation rate & high/low limiter processing. The primary loop can be tracked.	Page 254 M+P_2PIDH_T_

### ■2-degree-of-freedom Advanced PID Control (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+P_2PIDH_	Optimizes the responsive performance (tracking performance) in a setting value change and control performance to a disturbance, and outputs the result ( $\Delta MV$ ). This function block performs two-degree-of-freedom PID operation, PV tracking, integration stop, derivative stop, and SV variation rate & high/low limiter processing.	Page 262 M+P_2PIDH_

### ■Position Type PID Control (Enable Tracking for primary loop/Disable Tracking from secondary loop)

Function block symbol	Processing details	Reference
M+P_PIDP_T	Performs PID operations using process variable differential, inexact differential, and position type, and outputs the result. The primary loop can be tracked.	Page 269 M+P_PIDP_T

### ■Position Type PID Control (Disable Tracking for primary loop/Disable Tracking from secondary loop)

Function block symbol	Processing details	Reference
M+P_PIDP	Performs PID operations using process variable differential, inexact differential, and position type, and outputs the result.	Page 276 M+P_PIDP

### ■Position Type PID Control (Enable Tracking for primary loop/Enable Tracking from secondary loop)

Function block symbol	Processing details	Reference
M+P_PIDP_EX_T_	Performs PID operations using process variable differential, inexact differential, and position type, and outputs the result. It also allows manipulated value bumpless switching and tracking from the primary loop and secondary loop at a change of control mode.	Page 283 M+P_PIDP_EX_T_

### ■Position Type PID Control (Disable Tracking for primary loop/Enable Tracking from secondary loop)

Function block symbol	Processing details	Reference
M+P_PIDP_EX_	Performs PID operations using process variable differential, inexact differential, and position type, and outputs the result. It also allows manipulated value bumpless switching and tracking from the secondary loop at a change of control mode.	Page 291 M+P_PIDP_EX_

### ■Sample PI Control (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+P_SPI_T	Performs PI control during operating time (ST_) and outputs the result ( $\Delta MV$ ). This function block holds the output ( $\Delta MV = 0$ ) during hold time (HT). The primary loop can be tracked.	Page 299 M+P_SPI_T

### ■Sample PI Control (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+P_SPI	Performs PI control during operating time (ST_) and outputs the result ( $\Delta MV$ ). This function block holds the output ( $\Delta MV = 0$ ) during hold time (HT).	Page 305 M+P_SPI

### ■I-PD Control (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+P_IPD_T	Controls values to slowly respond to a change of the setting value (SV) without any shocks and outputs the result ( $\Delta MV$ ) by using the process variable (PV) in the proportional term and derivative term. The primary loop can be tracked.	Page 310 M+P_IPD_T

### ■I-PD Control (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+P_IPD	Controls values to slowly respond to a change of the setting value (SV) without any shocks and outputs the result ( $\Delta MV$ ) by using the process variable (PV) in the proportional term and derivative term.	Page 316 M+P_IPD

### ■Blend PI Control (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+P_BPI_T	This function block is used when the control amount vibrates in a short period but is stable in a long period. The primary loop can be tracked.	Page 321 M+P_BPI_T

### ■Blend PI Control (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+P_BPI	This function block is used when the control amount vibrates in a short period but is stable in a long period.	Page 326 M+P_BPI

### ■High/Low Limit Alarm Check

Function block symbol	Processing details	Reference
M+P_PHPL	Performs the high high limit/high limit/low limit/low low limit checks and the variation rate check to the input (PVPI) and outputs the results. If a value has exceeded an allowable range, an alarm occurs.	Page 331 M+P_PHPL

### ■2 position ON/OFF Control (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+P_ONF2_T	Performs two-position (on/off) control. The primary loop can be tracked.	Page 335 M+P_ONF2_T

## ■2 position ON/OFF Control (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+P_ONF2	Performs two-position (on/off) control.	Page 338 M+P_ONF2

## ■3 position ON/OFF Control (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+P_ONF3_T	Performs three-position (on/off) control. The primary loop can be tracked.	Page 341 M+P_ONF3_T

## ■3 position ON/OFF Control (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+P_ONF3	Performs three-position (on/off) control.	Page 345 M+P_ONF3

## ■Program Setter

Function block symbol	Processing details	Reference
M+P_PGS	Registers time width sets and setting value programs of up to 16 steps, and outputs a setting value corresponding to the progress time of each step with linear interpolation.	Page 348 M+P_PGS

## ■Multi-Point Program Setter

Function block symbol	Processing details	Reference
M+P_PGS2_	Registers time width sets and setting value programs of up to 32 steps, and outputs a setting value corresponding to the progress time of each step with linear interpolation.	Page 352 M+P_PGS2_

## ■Loop Selector (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+P_SEL	Selects an input value depending on the setting of the select signal and outputs the result. The input 1 and 2 are not tracked.	Page 363 M+P_SEL

## ■Loop Selector (Enable Tracking for primary loop only by CASIN\_T)

Function block symbol	Processing details	Reference
M+P_SEL_T1	Selects an input value depending on the setting of the select signal and outputs the result. The input 1 is not tracked. The primary loop of the input 2 is tracked, but tracking from the secondary loop is not performed.	Page 366 M+P_SEL_T1

## ■Loop Selector (Enable Tracking for primary loop by CASIN\_T1/\_T2)

Function block symbol	Processing details	Reference
M+P_SEL_T2	Selects an input value depending on the setting of the select signal and outputs the result. The primary loop of the input 1 and 2 is tracked, but tracking from the secondary loop is not performed.	Page 370 M+P_SEL_T2

## ■Loop Selector (Enable Tracking from secondary loop to primary loop)

Function block symbol	Processing details	Reference
M+P_SEL_T3_	Selects two input values depending on the setting of the select signal and outputs the result. The primary loop of the input 1 and 2 is tracked from the secondary loop.	Page 374 M+P_SEL_T3_



## Special

The following FB changes the control mode.

### ■Change Control Mode

Function block symbol	Processing details	Reference
M+P_MCHG	This function block switches the control mode to one of MANUAL, AUTO, CASCADE, COMPUTER MV, COMPUTER SV, and CASCADE DIRECT according to a mode change signal.	Page 378 M+P_MCHG

### ■Primary Loop Control Mode Switching

Function block symbol	Processing details	Reference
M+P_MCHGPRMRY	For the cascade connection, this function block switches the control mode of the primary loop to MANUAL when an alarm occurs or when the control mode is switched.	Page 381 M+P_MCHGPRMRY

## 4.3 Tag FBs

### Loop tag

The following FBs perform loop control processing including ratio control, various PID controls, two-position (on/off) control, three-position (on/off) control, program setter, and loop selector.

#### ■Velocity Type PID Control (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_PID_T	Performs velocity type PID control combining the functions of M+P_IN, M+P_PHPL, M+P_PID_T, and M+P_OUT1 into a single FB. The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.	Page 386 M+M_PID_T

#### ■Velocity Type PID Control (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_PID	Performs velocity type PID control combining the functions of M+P_IN, M+P_PHPL, M+P_PID, and M+P_OUT1 into a single FB.	Page 390 M+M_PID

#### ■Velocity Type PID Control and DUTY Output (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_PID_DUTY_T	Performs velocity type PID control and duty output combining the functions of M+P_IN, M+P_PHPL, M+P_PID_T, and M+P_DUTY into a single FB. The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.	Page 393 M+M_PID_DUTY_T

#### ■Velocity Type PID Control and DUTY Output (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_PID_DUTY	Performs velocity type PID control and duty output combining the functions of M+P_IN, M+P_PHPL, M+P_PID, and M+P_DUTY into a single FB.	Page 396 M+M_PID_DUTY

#### ■2-degree-of-freedom PID Control (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_2PID_T	Performs two-degree-of-freedom PID control combining the functions of M+P_IN, M+P_PHPL, M+P_2PID_T, and M+P_OUT1 into a single FB. The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.	Page 399 M+M_2PID_T

#### ■2-degree-of-freedom PID Control (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_2PID	Performs two-degree-of-freedom PID control combining the functions of M+P_IN, M+P_PHPL, M+P_2PID, and M+P_OUT1 into a single FB.	Page 402 M+M_2PID

#### ■2-degree-of-freedom PID Control and DUTY O/P (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_2PID_DUTY_T	Performs two-degree-of-freedom PID control and duty output combining the functions of M+P_IN, M+P_PHPL, M+P_2PID_T, and M+P_DUTY into a single FB. The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.	Page 405 M+M_2PID_DUTY_T

#### ■2-degree-of-freedom PID Control and DUTY O/P (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_2PID_DUTY	Performs two-degree-of-freedom PID control and duty output combining the functions of M+P_IN, M+P_PHPL, M+P_2PID, and M+P_DUTY into a single FB.	Page 408 M+M_2PID_DUTY

### ■2-degree-of-freedom Advanced PID Control (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_2PIDH_T_	Performs two-degree-of-freedom PID control combining the functions of M+P_IN, M+P_PHPL, M+P_2PIDH_T_, and M+P_OUT3_ into a single FB with PV/MV correction. The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, tag stop, or control mode change occurs, the primary loop control mode is switched to MANUAL.	Page 411 M+M_2PIDH_T_

### ■2-degree-of-freedom Advanced PID Control (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_2PIDH_	Performs two-degree-of-freedom PID control combining the functions of M+P_IN, M+P_PHPL, M+P_2PIDH_, and M+P_OUT3_ into a single FB with PV/MV correction.	Page 417 M+M_2PIDH_

### ■Position Type PID Control (Enable Tracking for primary loop/Disable Tracking from secondary loop)

Function block symbol	Processing details	Reference
M+M_PIDP_T	Performs position type PID control combining the functions of M+P_IN, M+P_PHPL, and M+P_PIDP_T into a single FB. The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.	Page 424 M+M_PIDP_T

### ■Position Type PID Control (Disable Tracking for primary loop/Disable Tracking from secondary loop)

Function block symbol	Processing details	Reference
M+M_PIDP	Performs position type PID control combining the functions of M+P_IN, M+P_PHPL, and M+P_PIDP into a single FB.	Page 427 M+M_PIDP

### ■Position Type PID Control (Enable Tracking for primary loop/Enable Tracking from secondary loop)

Function block symbol	Processing details	Reference
M+M_PIDP_EX_T_	Performs position type PID control combining the functions of M+P_IN, M+P_PHPL, and M+P_PIDP_EX_T_ into a single FB. It also allows manipulated value bumpless switching at a change of control mode. In the cascade connection, it allows tracking of the primary loop or tracking from the secondary loop. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.	Page 430 M+M_PIDP_EX_T_

### ■Position Type PID Control (Disable Tracking for primary loop/Enable Tracking from secondary loop)

Function block symbol	Processing details	Reference
M+M_PIDP_EX_	Performs position type PID control combining the functions of M+P_IN, M+P_PHPL, and M+P_PIDP_EX_ into a single FB. It also allows manipulated value bumpless switching at a change of control mode. In the cascade connection, it allows tracking from the secondary loop.	Page 433 M+M_PIDP_EX_

### ■Sample PI Control (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_SPI_T	Performs sample PI control combining the functions of M+P_IN, M+P_PHPL, M+P_SPI_T, and M+P_OUT1 into a single FB. The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.	Page 436 M+M_SPI_T

### ■Sample PI Control (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_SPI	Performs sample PI control combining the functions of M+P_IN, M+P_PHPL, M+P_SPI, and M+P_OUT1 into a single FB.	Page 439 M+M_SPI

### ■I-PD Control (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_IPD_T	Performs I-PD control combining the functions of M+P_IN, M+P_PHPL, M+P_IPD_T, and M+P_OUT1 into a single FB. The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.	Page 442 M+M_IPD_T

### ■I-PD Control (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_IPD	Performs I-PD control combining the functions of M+P_IN, M+P_PHPL, M+P_IPD, and M+P_OUT1 into a single FB.	Page 445 M+M_IPD

### ■Blend PI Control (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_BPI_T	Performs blend PI control combining the functions of M+P_IN, M+P_PHPL, M+P_BPI_T, and M+P_OUT1 into a single FB. The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.	Page 448 M+M_BPI_T

### ■Blend PI Control (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_BPI	Performs blend PI control combining the functions of M+P_IN, M+P_PHPL, M+P_BPI, and M+P_OUT1 into a single FB.	Page 451 M+M_BPI

### ■Ratio Control (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_R_T	Performs ratio control combining the functions of M+P_IN, M+P_PHPL, M+P_R_T, and M+P_OUT2 into a single FB. The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.	Page 454 M+M_R_T

### ■Ratio Control (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_R	Performs ratio control combining the functions of M+P_IN, M+P_PHPL, M+P_R, and M+P_OUT2 into a single FB.	Page 457 M+M_R

### ■2 position ON/OFF Control (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_ONF2_T	Performs two-position (on/off) control combining the functions of M+P_IN, M+P_PHPL, and M+P_ONF2_T into a single FB. The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.	Page 460 M+M_ONF2_T

### ■2 position ON/OFF Control (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_ONF2	Performs two-position (on/off) control combining the functions of M+P_IN, M+P_PHPL, and M+P_ONF2 into a single FB.	Page 463 M+M_ONF2

### ■3 position ON/OFF Control (Enable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_ONF3_T	Performs three-position (on/off) control combining the functions of M+P_IN, M+P_PHPL, and M+P_ONF3_T into a single FB. The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.	Page 466 M+M_ONF3_T

### ■3 position ON/OFF Control (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_ONF3	Performs three-position (on/off) control combining the functions of M+P_IN, M+P_PHPL, and M+P_ONF3 into a single FB.	Page 469 M+M_ONF3

## ■Monitor

Function block symbol	Processing details	Reference
M+M_MONI	Performs monitoring combining the functions of M+P_IN and M+P_PHPL into a single FB.	Page 472 M+M_MONI

## ■Manual Output with Monitor

Function block symbol	Processing details	Reference
M+M_MWM	Performs manual output with monitor combining the functions of M+P_IN, M+P_PHPL, and M+P_MOUT into a single FB.	Page 474 M+M_MWM

## ■Batch Preparation

Function block symbol	Processing details	Reference
M+M_BC	Performs batch preparation combining the functions of M+P_PSUM and M+P_BC into a single FB.	Page 476 M+M_BC

## ■Pulse Integrator

Function block symbol	Processing details	Reference
M+M_PSUM	Performs pulse integration combining the functions of M+P_PSUM into a single FB.	Page 478 M+M_PSUM

## ■Loop Selector (Disable Tracking for primary loop)

Function block symbol	Processing details	Reference
M+M_SEL	Performs loop selector combining the functions of M+P_SEL into a single FB.	Page 480 M+M_SEL

## ■Loop Selector (Enable Tracking for primary loop only by CASIN\_T)

Function block symbol	Processing details	Reference
M+M_SEL_T1	Performs loop selector combining the functions of M+P_SEL_T1 into a single FB. The input 1 is not tracked in the cascade connection. The primary loop of the input 2 is tracked, but tracking from the secondary loop is not performed. If output open, stop alarm, or control mode change occurs, the control mode of the primary loop of the input 2 is switched to MANUAL.	Page 482 M+M_SEL_T1

## ■Loop Selector (Enable Tracking for primary loop by CASIN\_T1/\_T2)

Function block symbol	Processing details	Reference
M+M_SEL_T2	Performs loop selector combining the functions of M+P_SEL_T2 into a single FB. In the cascade connection, the primary loop of the inputs 1 and 2 is tracked, but tracking from the secondary loop is not performed. If output open, stop alarm, or control mode change occurs, the control mode of the primary loop of the inputs 1 and 2 is switched to MANUAL.	Page 484 M+M_SEL_T2

## ■Loop Selector (Enable Tracking from secondary loop to primary loop)

Function block symbol	Processing details	Reference
M+M_SEL_T3_	Performs loop selector combining the functions of M+P_SEL_T3_ into a single FB. In the cascade connection, performs loop selector combining the functions of M+P_SEL_T3_ into a single FB. The primary loop of the input 1 and 2 can be tracked from the secondary loop. If output open, stop alarm, or control mode change occurs, the control mode of the primary loop of the inputs 1 and 2 is switched to MANUAL.	Page 487 M+M_SEL_T3_

## ■Manual Output

Function block symbol	Processing details	Reference
M+P_MOUT	Performs manual output combining the functions of M+P_MOUT into a single FB.	Page 490 M+M_MOUT

## ■Program Setter

Function block symbol	Processing details	Reference
M+M_PGS	Performs program settings combining the functions of M+P_PGS into a single FB.	Page 492 M+M_PGS

## ■Multi-Point Program Setter

Function block symbol	Processing details	Reference
M+M_PGS2_	Performs program settings combining the functions of M+P_PGS2_ into a single FB. Multiple FBs can be connected to execute time width sets and setting value programs of 32 steps or more.	Page 494 M+M_PGS2_

## ■ Manual Setter with Monitor

Function block symbol	Processing details	Reference
M+M_SWM_	Performs manual settings with monitor combining the functions of M+P_IN, M+P_PHPL, and M+P_MSET_ into a single FB.	Page 498 M+M_SWM_

## ■ Position Proportional Output

Function block symbol	Processing details	Reference
M+M_PVAL_T_	Outputs open/close command bits to operate the motor valve opening according to the deviation of opening between the feedback input and setting value. The primary loop can be tracked in the cascade connection. If a sensor error, stop alarm, tag stop, or control mode change occurs, the primary loop control mode is switched to MANUAL.	Page 501 M+M_PVAL_T_

## ■ Heating and Cooling Output

Function block symbol	Processing details	Reference
M+M_HTCL_T_	Outputs two manipulated values after split conversion and output conversion to setting values. This function block can perform temperature control by outputting the values to both the heating and cooling operation terminal. The primary loop can be tracked in the cascade connection. If output open, stop alarm, tag stop, or control mode change occurs, the primary loop control mode is switched to MANUAL.	Page 510 M+M_HTCL_T_

## Status tag

The following FBs perform reversible/irreversible operation and ON/OFF operation and operates as a timer and a counter.

### ■Motor Irreversible (2 Input/2 Output)

Function block symbol	Processing details	Reference
M+M_NREV	Performs irreversible operation and controls a solenoid valve.	Page 520 M+M_NREV

### ■Motor Reversible (2 Input/3 Output)

Function block symbol	Processing details	Reference
M+M_REV	Performs reversible operation.	Page 524 M+M_REV

### ■ON/OFF Operation (2 Input/2 Output)

Function block symbol	Processing details	Reference
M+M_MVAL1	Controls an ON/OFF motor valve and solenoid valve.	Page 528 M+M_MVAL1

### ■ON/OFF Operation (2 Input/3 Output)

Function block symbol	Processing details	Reference
M+M_MVAL2	Controls an ON/OFF motor valve (with intermediate status).	Page 532 M+M_MVAL2

### ■Timer 1 (Timer stops when COMPLETE flag is ON)

Function block symbol	Processing details	Reference
M+M_TIMER1	This function block is a clock timer. It stops counting the time when the timer count value reaches the set value.	Page 536 M+M_TIMER1

### ■Timer 2 (Timer continues when COMPLETE flag is ON)

Function block symbol	Processing details	Reference
M+M_TIMER2	This function block is a clock timer. It continues to count the time even if the time count value exceeds the set value, and the timer stops when the time count value reaches the timer high limit value.	Page 538 M+M_TIMER2

### ■Counter 1 (Counter stops when COMPLETE flag is ON)

Function block symbol	Processing details	Reference
M+M_COUNTER1	This function block is a counter that counts contact signal inputs. It stops counting the inputs when the count value reaches the set value.	Page 540 M+M_COUNTER1

### ■Counter 2 (Counter continues when COMPLETE flag is ON)

Function block symbol	Processing details	Reference
M+M_COUNTER2	This function block is a counter that counts contact signal inputs. It continues to count the inputs even if the count value exceeds the set value, and stops counting the input when the count value reaches the counter high limit value.	Page 543 M+M_COUNTER2

### ■Push Button Operation (5 Input/5 Output)

Function block symbol	Processing details	Reference
M+M_PB_	Performs push button operation.	Page 546 M+M_PB_

## Alarm tag

The following FBs perform alarm notification.

### ■Alarm

Function block symbol	Processing details	Reference
M+M_ALARM	Displays the alarms for the input pins ALMIN1 to ALMIN8 to which TRUE is input on the "Alarm List" window of the PX Developer monitor tool.	Page 550 M+M_ALARM

### ■64 Points Alarm

Function block symbol	Processing details	Reference
M+M_ALARM_64PT_	Displays the alarms for the bits of the input pins ALMIN_W1 to ALMIN_W4 to which TRUE is input on the "Alarm List" window of the PX Developer monitor tool.	Page 552 M+M_ALARM_64PT_

## Message tag

The following FBs perform message notification.

### ■Message

Function block symbol	Processing details	Reference
M+M_MESSAGE	Displays the events for the input pins MSGIN1 to MSGIN8 to which TRUE is input on the "Event List" window of the PX Developer monitor tool.	Page 554 M+M_MESSAGE

### ■64 Points Message

Function block symbol	Processing details	Reference
M+M_MESSAGE_64PT_	Displays the events for the bits of the input pins MSGIN_W1 to MSGIN_W4 to which TRUE is input on the "Event List" window of the PX Developer monitor tool.	Page 556 M+M_MESSAGE_64P T_



This part consists of the following chapters.

5 ANALOG VALUE SELECTION AND AVERAGE VALUE

---

6 CORRECTION OPERATION

---

7 ARITHMETIC OPERATION

---

8 COMPARISON OPERATION

---

9 CONTROL OPERATION

---

# 5 ANALOG VALUE SELECTION AND AVERAGE VALUE

The following FBs output the maximum, minimum, middle, average, and absolute values of input values.

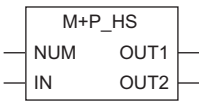
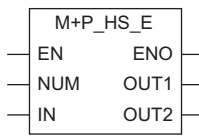
**Point** 

The output variables and public variables of general process FBs are latched.

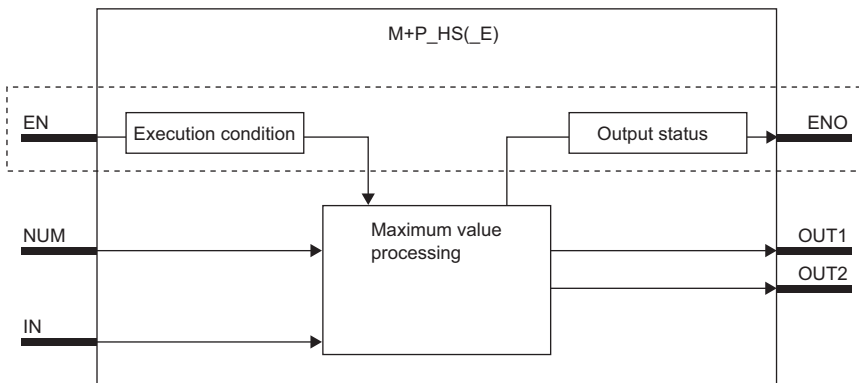
## 5.1 High Selector (M+P\_HS(\_E))

### M+P\_HS(\_E)

These FBs output the maximum value of input values (array).

FBD/LD	
[Without EN/ENO] 	[With EN/ENO] 

#### ■Block diagram



Processing in the dashed part is only carried out in functions with EN/ENO pins.

#### Setting data

#### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
EN	Execution condition	TRUE: Execute FALSE: Stop	Input variable	BOOL
NUM	Input numbers	1 to 16	Input variable	INT
IN	Input 1 to Input 16	-999999 to 999999	Input variable	REAL array element <sup>*1</sup> (Number of elements: 16)
ENO	Output status	TRUE: Normal FALSE: Abnormal or operation stop	Output variable	BOOL
OUT1	Output	-999999 to 999999	Output variable	REAL
OUT2	Output selection	0H to FFFFH	Output variable	WORD

\*1 Even when an input number to be used is less than 16, specify an array whose data type is REAL and the number of elements is 16. When an input number is less than 16, input values must be left-justified from the start element of the array.

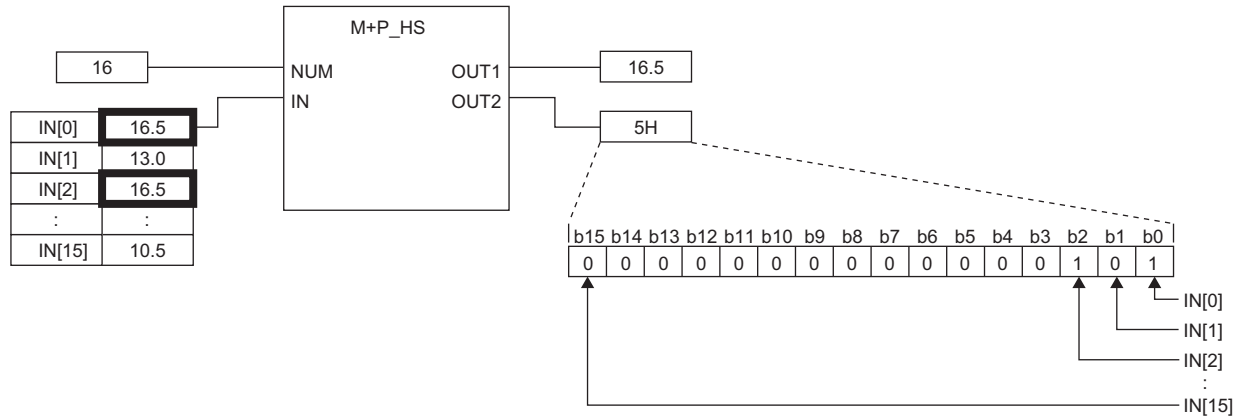
## Processing details

### ■ Operation processing

These functions output the maximum value of the values input to the input variables IN[0] to IN[15] from the output variable OUT1. They set the bit corresponding to the input value selected as the maximum value from the input variables IN[0] to IN[15] specified by the input number NUM to 1 and output the bit from the output variable OUT2. (When multiple maximum values are input, they set all the corresponding bits to 1 and output the bits from the output variable OUT2.)

**Ex.**

When the input values of the input variables IN[0] and IN[2] are the maximum values



### ■ Operation result

- Function block without EN/ENO

Operation result	OUT1, OUT2
No operation error	Operation result output value
Operation error	Undefined value

- Function block with EN/ENO

The execution conditions and operation results will be as follows.

Execution condition	Operation result	
	ENO	OUT1, OUT2
TRUE (operation execution)	TRUE (no operation error)	Operation result output value
	FALSE (operation error)*1	Undefined value
FALSE (operation stop)	FALSE*1	Undefined value

\*1 If the value FALSE is output from ENO, the output data from this function block will be undefined. Create a program so that the undefined value will not be used in operations.

## Operation error

Error code (SD0)	Description
3402H	Input 1 to 16 are subnormal numbers or NaN (not a number).
3405H	The number of inputs is less than 1 or greater than 16.

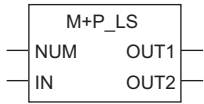
## 5.2 Low Selector (M+P\_LS(\_E))

### M+P\_LS(\_E)

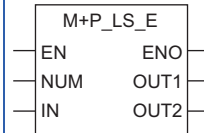
These FBs output the minimum value of input values (array).

#### FBD/LD

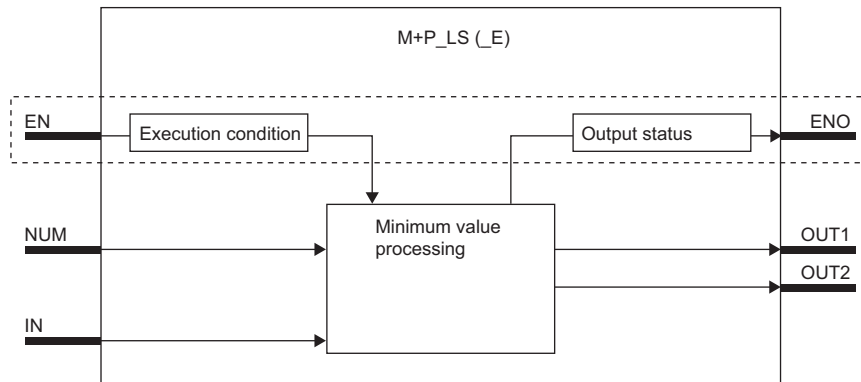
[Without EN/ENO]



[With EN/ENO]



#### ■Block diagram



Processing in the dashed part is only carried out in functions with EN/ENO pins.

#### Setting data

#### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
EN	Execution condition	TRUE: Execute FALSE: Stop	Input variable	BOOL
NUM	Input numbers	1 to 16	Input variable	INT
IN	Input 1 to Input 16	-999999 to 999999	Input variable	REAL array element <sup>*1</sup> (Number of elements: 16)
ENO	Output status	TRUE: Normal FALSE: Abnormal or operation stop	Output variable	BOOL
OUT1	Output	-999999 to 999999	Output variable	REAL
OUT2	Output selection	0H to FFFFH	Output variable	WORD

\*1 Even when an input number to be used is less than 16, specify an array whose data type is REAL and the number of elements is 16. When an input number is less than 16, input values must be left-justified from the start element of the array.

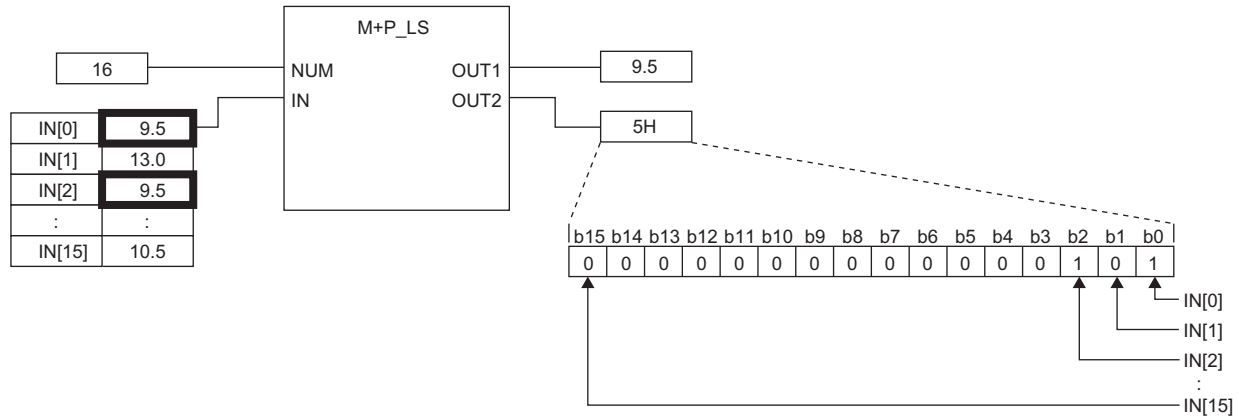
## Processing details

### ■ Operation processing

These functions output the minimum value of the values input to the input variables IN[0] to IN[15] from the output variable OUT1. They set the bit corresponding to the input value selected as the minimum value from the input variables IN[0] to IN[15] specified by the input number NUM to 1 and output the bit from the output variable OUT2. (When multiple minimum values are input, they set all the corresponding bits to 1 and output the bits from the output variable OUT2.)

**Ex.**

When the input values of the input variables IN[0] and IN[2] are the minimum values



5

### ■ Operation result

- Function block without EN/ENO

Operation result	OUT1, OUT2
No operation error	Operation result output value
Operation error	Undefined value

- Function block with EN/ENO

The execution conditions and operation results will be as follows.

Execution condition	Operation result	
	ENO	OUT1, OUT2
TRUE (operation execution)	TRUE (no operation error)	Operation result output value
	FALSE (operation error) <sup>*1</sup>	Undefined value
FALSE (operation stop)	FALSE <sup>*1</sup>	Undefined value

\*1 If the value FALSE is output from ENO, the output data from this function block will be undefined. Create a program so that the undefined value will not be used in operations.

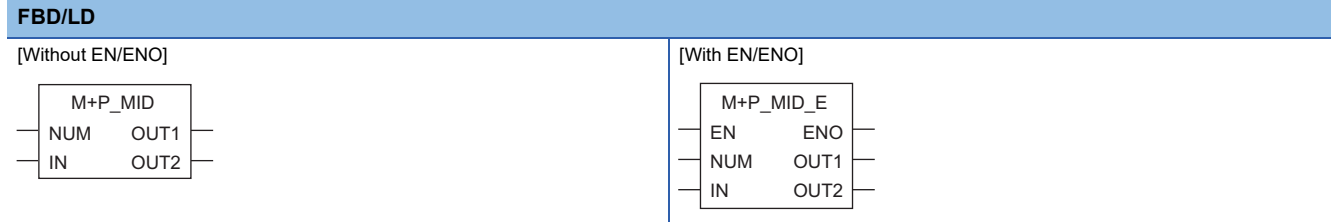
## Operation error

Error code (SD0)	Description
3402H	Input 1 to 16 are subnormal numbers or NaN (not a number).
3405H	The number of inputs is less than 1 or greater than 16.

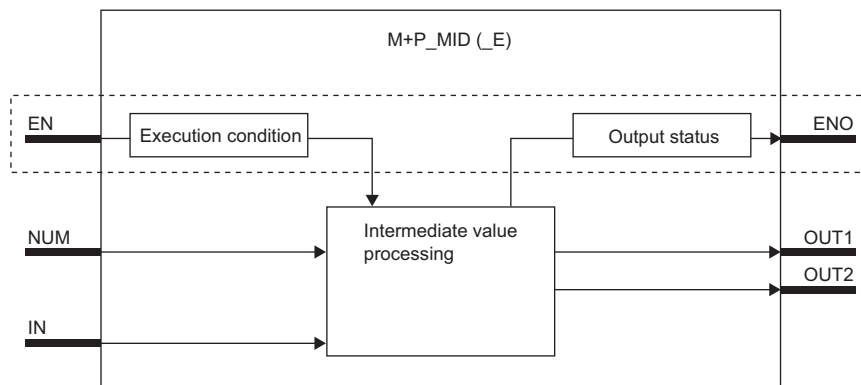
## 5.3 Middle Value Selection (M+P\_MID(\_E))

### M+P\_MID(\_E)

These FBs output the middle value of input values (array).



#### ■Block diagram



Processing in the dashed part is only carried out in functions with EN/ENO pins.

#### Setting data

#### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
EN	Execution condition	TRUE: Execute FALSE: Stop	Input variable	BOOL
NUM	Input numbers	1 to 16	Input variable	INT
IN	Input 1 to Input 16	-999999 to 999999	Input variable	REAL array element <sup>*1</sup> (Number of elements: 16)
ENO	Output status	TRUE: Normal FALSE: Abnormal or operation stop	Output variable	BOOL
OUT1	Output	-999999 to 999999	Output variable	REAL
OUT2	Output selection	0H to FFFFH	Output variable	WORD

\*1 Even when an input number to be used is less than 16, specify an array whose data type is REAL and the number of elements is 16. When an input number is less than 16, input values must be left-justified from the start element of the array.

## Processing details

### ■ Operation processing

These functions output the intermediate value of the values input to the input variables IN[0] to IN[15] from the output variable OUT1.

- They rearrange the input values in order from the smallest one to the largest one, and output the intermediate value of the input values.

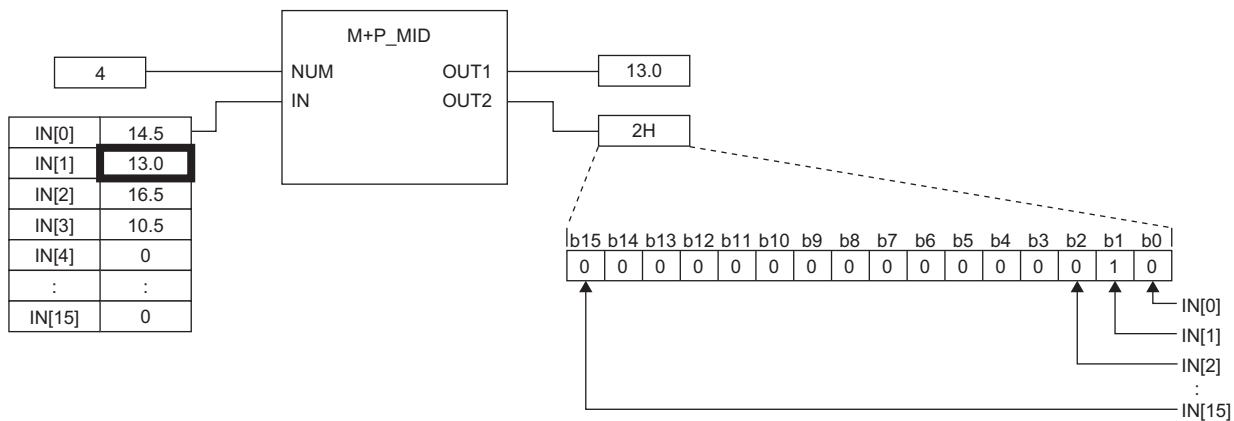
[Example] 1, 3, 4, 5, 1 → 1, 1, 3, 4, 5  
 In this case, 3 is output as the intermediate value.

- If the number of inputs is even, they output the smaller one from the output variable OUT1.

[Example] 1, 3, 4, 5, 1, 4 → 1, 1, 3, 4, 4, 5  
 In this case, 3 is output as the intermediate value.

They set the bit corresponding to the input value selected as the intermediate value from the input variables IN[0] to IN[15] specified by the input number NUM to 1 and output the bit from the output variable OUT2. (When multiple intermediate values are input, they set all the corresponding bits to 1 and output the bits from the output variable OUT2.)

[Example] When the intermediate value of the input variables IN[0] to IN[3] is output



### ■ Operation result

- Function block without EN/ENO

Operation result	OUT1, OUT2
No operation error	Operation result output value
Operation error	Undefined value

- Function block with EN/ENO

The execution conditions and operation results will be as follows.

Execution condition	Operation result	
	ENO	OUT1, OUT2
TRUE (operation execution)	TRUE (no operation error)	Operation result output value
	FALSE (operation error)*1	Undefined value
FALSE (operation stop)	FALSE*1	Undefined value

\*1 If the value FALSE is output from ENO, the output data from this function block will be undefined. Create a program so that the undefined value will not be used in operations.

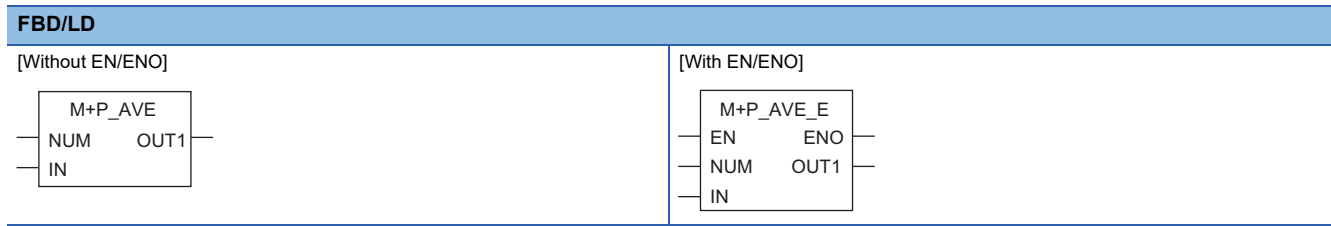
## Operation error

Error code (SD0)	Description
3402H	Input 1 to 16 are subnormal numbers or NaN (not a number).
3405H	The number of inputs is less than 1 or greater than 16.

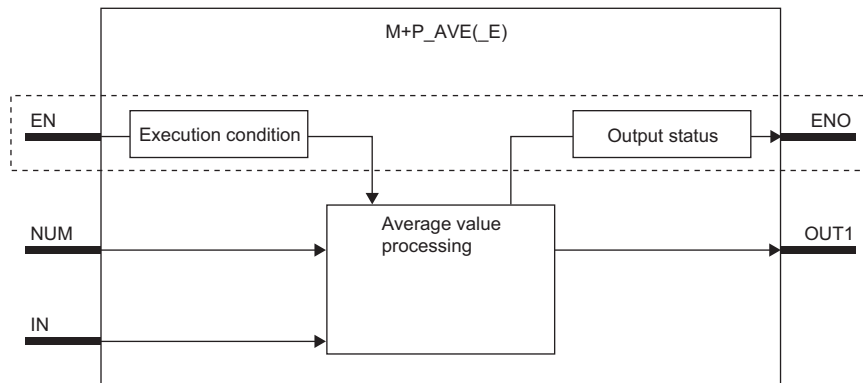
# 5.4 Average Value (M+P\_AVE(\_E))

## M+P\_AVE(\_E)

These FBs output the average value of input values (array).



### Block diagram



Processing in the dashed part is only carried out in functions with EN/ENO pins.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
EN	Execution condition	TRUE: Execute FALSE: Stop	Input variable	BOOL
NUM	Input numbers	1 to 16	Input variable	INT
IN	Input 1 to Input 16	-999999 to 999999	Input variable	REAL array element <sup>*1</sup> (Number of elements: 16)
ENO	Output status	TRUE: Normal FALSE: Abnormal or operation stop	Output variable	BOOL
OUT1	Output	-999999 to 999999	Output variable	REAL

\*1 Even when an input number to be used is less than 16, specify an array whose data type is REAL and the number of elements is 16. When an input number is less than 16, input values must be left-justified from the start element of the array.



## Processing details

### ■ Operation processing

These functions output the mean value of the values input to the input variables IN[0] to IN[15] from the output variable OUT1.

$$OUT = (IN[0] + IN[1] + IN[2] + \dots + IN[15]) \div NUM$$

IN[0] to IN[15]: Input value, OUT1: Output value, NUM: Input number

### ■ Operation result

- Function block without EN/ENO

Operation result	OUT1, OUT2
No operation error	Operation result output value
Operation error	Undefined value

- Function block with EN/ENO

The execution conditions and operation results will be as follows.

Execution condition	Operation result	
EN	ENO	OUT1, OUT2
TRUE (operation execution)	TRUE (no operation error)	Operation result output value
	FALSE (operation error)*1	Undefined value
FALSE (operation stop)	FALSE*1	Undefined value

\*1 If the value FALSE is output from ENO, the output data from this function block will be undefined. Create a program so that the undefined value will not be used in operations.

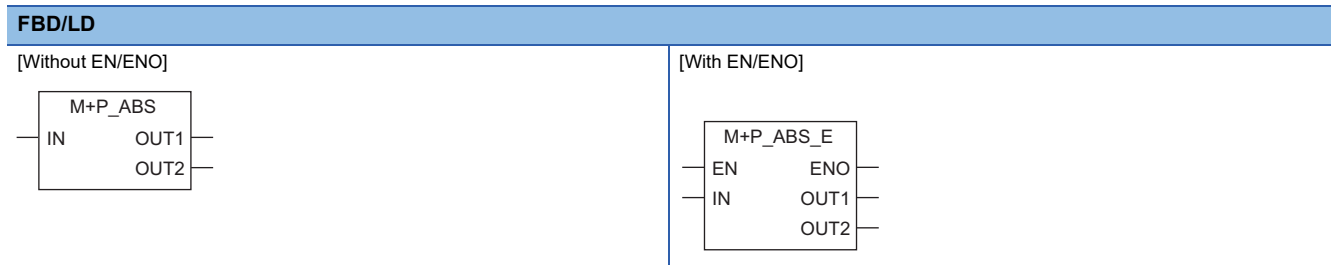
## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input 1 to 16 are subnormal numbers or NaN (not a number).
3403H	An overflow has occurred.
3405H	The number of inputs is less than 1 or greater than 16.

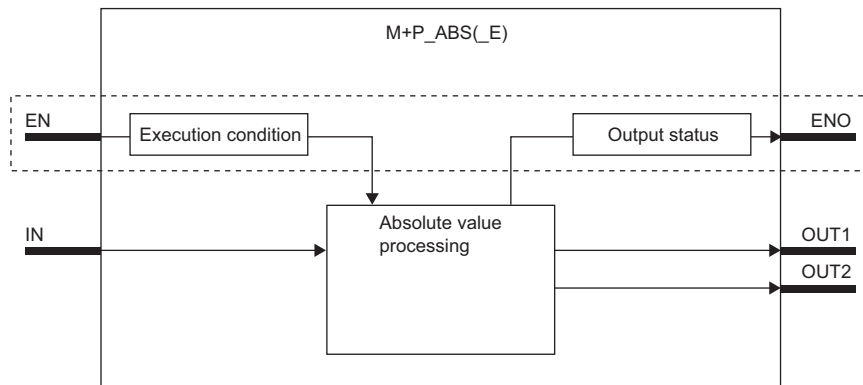
# 5.5 Absolute Value (M+P\_ABS(\_E))

## M+P\_ABS(\_E)

These FBs output the absolute value of an input value.



### ■Block diagram



Processing in the dashed part is only carried out in functions with EN/ENO pins.

### Setting data

#### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
EN	Execution condition	TRUE: Execute FALSE: Stop	Input variable	BOOL
IN	Input	-999999 to 999999	Input variable	REAL
ENO	Output status	TRUE: Normal FALSE: Abnormal or operation stop	Output variable	BOOL
OUT1	Output	-999999 to 999999	Output variable	REAL
OUT2	Input value sign detection	0H to 2H 0H: IN = 0 1H: IN > 0 2H: IN < 0	Output variable	WORD

## Processing details

### ■ Operation processing

These functions output the absolute value of a value input to the input variable IN from the output variable OUT1, and output the sign determination of the input value from the output variable OUT2.

- Processing of OUT1

Input value (IN)	Processing	Output value (OUT1)
	$OUT1 =  IN $	

- Processing of OUT2

Input value (IN)	Processing	Output value (OUT2)
	$IN = 0: OUT2 = 0H$ $IN > 0: OUT2 = 1H$ $IN < 0: OUT2 = 2H$	

### ■ Operation result

- Function block without EN/ENO

Operation result	OUT1, OUT2
No operation error	Operation result output value
Operation error	Undefined value

- Function block with EN/ENO

The execution conditions and operation results will be as follows.

Execution condition	Operation result	
	ENO	OUT1, OUT2
TRUE (operation execution)	TRUE (no operation error)	Operation result output value
	FALSE (operation error) <sup>*1</sup>	Undefined value
FALSE (operation stop)	FALSE <sup>*1</sup>	Undefined value

\*1 If the value FALSE is output from ENO, the output data from this function block will be undefined. Create a program so that the undefined value will not be used in operations.

## Operation error

Error code (SD0)	Description
3402H	Input 1 to 16 are subnormal numbers or NaN (not a number).

# 6 CORRECTION OPERATION


The following FBs perform operation processing such as broken line correction, standard filter, engineering value conversion, temperature/pressure correction, and summation.

## Point

The output variables and public variables of general process FBs are latched.

## Point

To set initial values of public variables of the general process FB arranged on a user-defined tag FB in "FB Property" of the engineering tool, refer to the following.

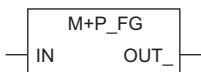
 GX Works3 Operating Manual

## 6.1 Function Generator (M+P\_FG)

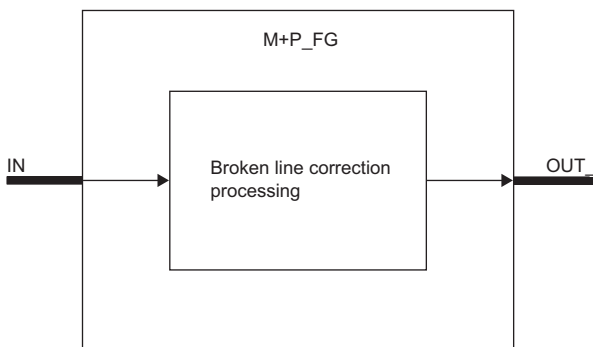
### M+P\_FG

This FB outputs (OUT\_) values according to the function generator pattern that consists of SN break points with regard to the input (IN).

FBD/LD



### Block diagram



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN	Input	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL

#### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
SN	Number of Points	0 to 48	0	User	INT
X1_ to X48_	Input coordinates (X Coordinates)	-999999 to 999999	0.0	User	REAL
Y1_ to Y48_	Output coordinates (Y Coordinates)	-999999 to 999999	0.0	User	REAL

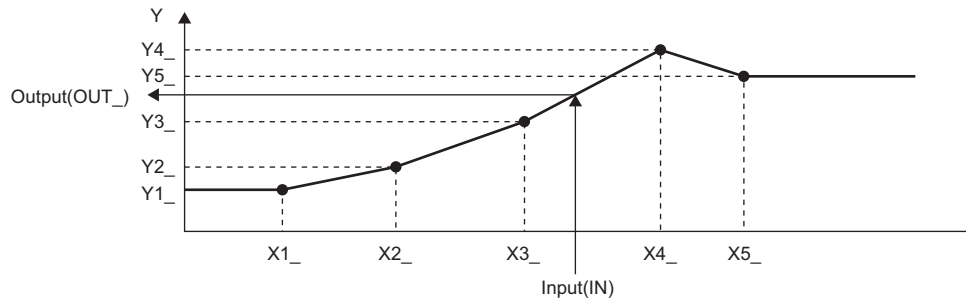
## Processing details

### ■ Broken line correction processing

This function block outputs values according to a function generator pattern that consists of SN (the number of break points = 0 to 48) from the output variable OUT\_ with regard to the values input from the input variable IN.

**Ex.**

When the number of break points is 5 (SN = 5)



#### • Processing details

This function block performs the following operations.

Input (IN)	Output (OUT_)
$IN \leq X1_$	$OUT_ = Y1_$
$X_{i-1}_ < IN \leq X_i_ (i = 2 \text{ to } n)$	$OUT_ = \frac{Y_i_ - Y_{i-1}_}{X_i_ - X_{i-1}_} \times (IN - X_{i-1}_) + Y_{i-1}_$
$IN > Xn_$	$OUT_ = Yn_$

$X_i_$ : Input coordinates

$Y_i_$ : Output coordinates

IN: Input value

OUT\_: Output value

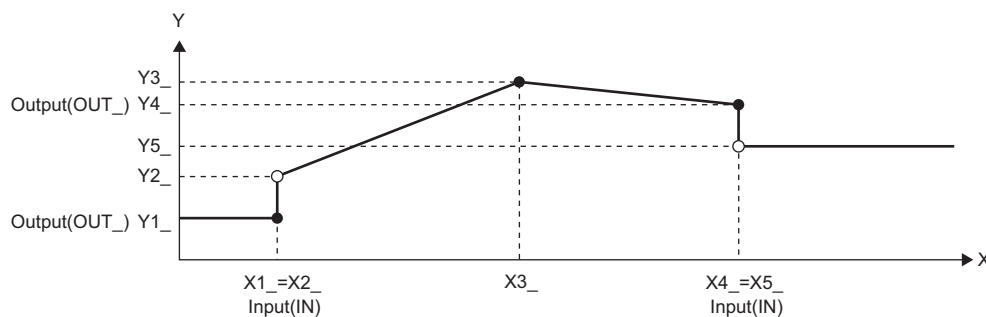
- Set the input coordinates ( $X_i_$ ) as follows:  $X_i_ \leq X_{i+1}_$ . (For  $X_i_ > X_{i+1}_$ , the values to  $X_i_$  are valid as break points.)
- If multiple  $Y_i_$  coordinates exist for a single  $X_i_$  coordinate, the  $Y_i_$  coordinate with the smaller  $X_i_$  coordinate is applied.

**Ex.**

$X1_ = X2_$ ,  $X4_ = X5_$

When the input coordinate ( $X1_ = X2_$ ) is input to the input (IN), the output coordinate ( $Y1_$ ) is output.

When the input coordinate ( $X4_ = X5_$ ) is input to the input (IN), the output coordinate ( $Y4_$ ) is output.



#### • Relation between the range of the number of break points (SN) and processing

Number of break points (SN)	Processing
SN = 0	No processing is performed.
$0 < SN \leq 48$	Broken line correction processing is performed.
$SN < 0$ or $48 < SN$	An operation error occurs.

## Operation error

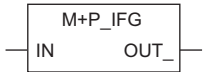
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (IN) or operation constant (X1_ to X48_, Y1_ to Y48_) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The number of break points (SN) is less than 0 or greater than 48.

## 6.2 Inverse Function Generator (M+P\_IFG)

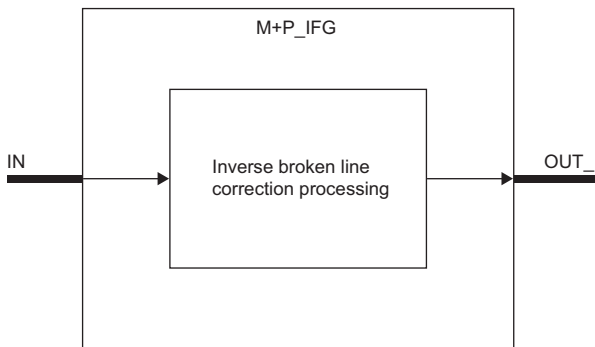
### M+P\_IFG

This FB outputs (OUT\_) values according to the function generator pattern that consists of SN break points with regard to the input (IN).

FBD/LD



### ■Block diagram



6

### Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN	Input	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL

### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
SN	Number of Points	0 to 48	0	User	INT
X1_ to X48_	Output coordinate (X coordinate)	-999999 to 999999	0.0	User	REAL
Y1_ to Y48_	Input coordinate (Y coordinate)	-999999 to 999999	0.0	User	REAL

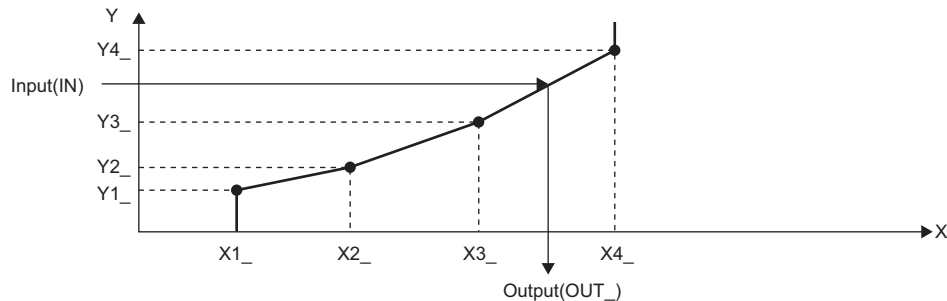
## Processing details

### ■ Inverse broken line correction

This function block outputs values according to a function generator pattern that consists of SN (the number of break points = 0 to 48) from the output variable OUT\_ with regard to the values input from the input variable IN.

**Ex.**

When the number of break points is 4 (SN = 4)



#### • Processing details

Input (IN)	Output (OUT_)
$IN \leq Y1_$	$OUT_ = X1_$
$Y_{i-1_} < IN \leq Y_{i_}$ ( $i = 2$ to $n$ )	$OUT_ = \frac{X_{i_} - X_{i-1_}}{Y_{i_} - Y_{i-1_}} \times (IN - Y_{i-1_}) + X_{i-1_}$
$IN > Y_{n_}$	$OUT_ = X_{n_}$

$X_{i_}$ : Output coordinates

$Y_{i_}$ : Input coordinates

IN: Input value

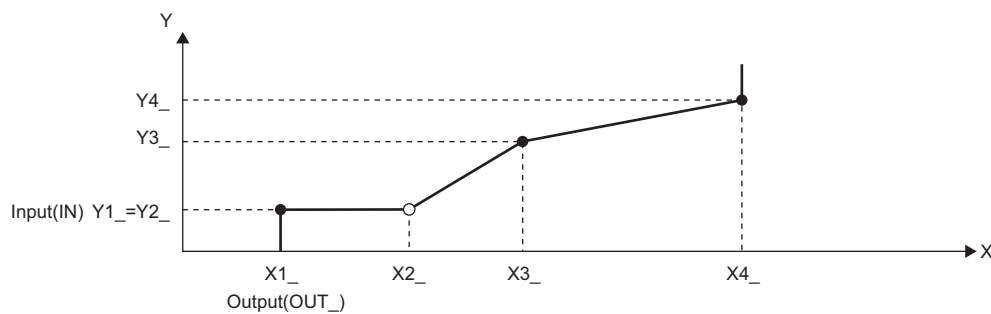
OUT\_: Output value

- Set the input coordinates ( $Y_{i_}$ ) as follows:  $Y_{i_} \leq Y_{i+1_}$ . (For  $Y_{i_} > Y_{i+1_}$ , the values to  $Y_{i_}$  are valid as break points.)
- If multiple  $X_{i_}$  coordinates exist for a single  $Y_{i_}$  coordinate, the  $X_{i_}$  coordinate with the smaller  $Y_{i_}$  coordinate is applied.

**Ex.**

$Y1_ = Y2_$

When the input coordinate ( $Y1_ = Y2_$ ) is input to the input (IN), the output coordinate ( $X1_$ ) is output.



#### • Relation between the range of the number of break points (SN) and processing

Number of break points (SN)	Processing
SN = 0	No processing is performed.
$0 < SN \leq 48$	Broken line correction processing is performed.
$SN < 0$ or $48 < SN$	An operation error occurs.



## Operation error

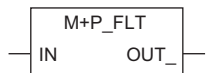
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (IN) or operation constant (X1_ to X48_, Y1_ to Y48_) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The number of break points (SN) is less than 0 or greater than 48.

## 6.3 Standard Filter (moving average) (M+P\_FLT)

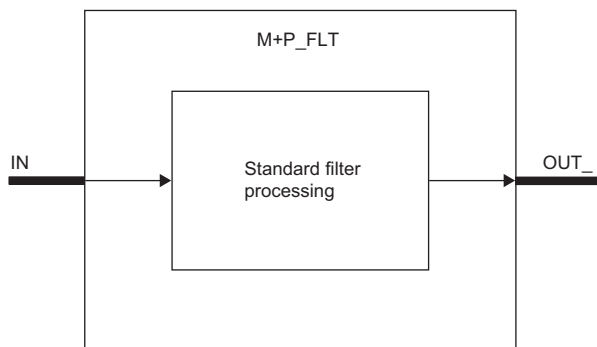
### M+P\_FLT

This FB outputs (OUT\_) the average value of SN pieces of data of the input (IN) collected the at data collection interval ST.

FBD/LD



#### ■Block diagram



#### Setting data

#### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN	Input	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL

#### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
ST_	Data Collection Interval	0 to 999[s]	1.0	User	REAL
SN	Sampling Count	0 to 48	0	User	INT

## Processing details

### ■ Standard filter processing

This function block outputs the mean value of SN pieces of data of the input (IN) value collected at the data collection interval ST\_ (s) from the output variable OUT\_.

This function block performs the following operations.

$$OUT\_ = \frac{IN1+IN2+IN3+ \dots +IN_{SN}}{SN}$$

SN: Sampling count, IN1 to IN<sub>SN</sub>: Input value, OUT\_: Output value

The data is refreshed in the following cycle. (The resultant value is rounded off to the nearest whole number.)

$$\frac{ST\_}{\Delta T}$$

ΔT: Execution cycle

Until the input (IN) reaches the sampling count (SN), the mean value of the input (IN) values that have been collected so far is output (OUT\_).

Set the data collection interval (ST\_) as follows: ST\_ = n × ΔT. (n is an integer.)

The following table lists the relation between the range of the sampling count (SN) and processing.

Sampling count (SN)	Processing
0 < SN ≤ 48	The standard filter processing is performed.
SN = 0	Output (OUT_) = 0
SN < 0 or 48 < SN	An operation error occurs.

## Operation error

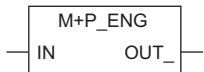
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (IN) or operation constant (ST_) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The sampling count (SN) is less than 0 or greater than 48.
	The execution cycle (ΔT) setting is less than 0.
	The data sampling interval (ST_) is less than 0.
	(Data sampling interval (ST_) ÷ execution cycle (ΔT)) is greater than 32767.

## 6.4 Engineering Value Conversion (M+P\_ENG)

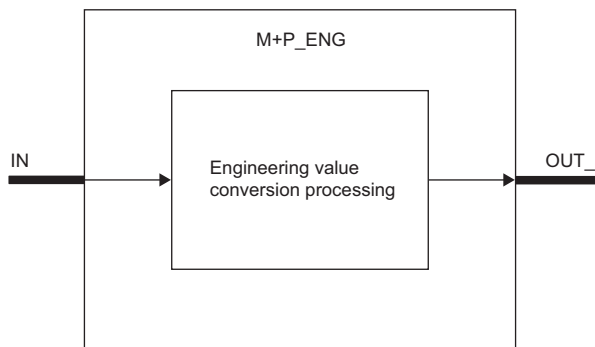
### M+P\_ENG

This FB converts a value (%) of the input (IN) in percentage into an engineering value such as temperature and pressure and outputs (OUT\_) the result.

FBD/LD



#### ■Block diagram



#### Setting data

#### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN	Input	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL

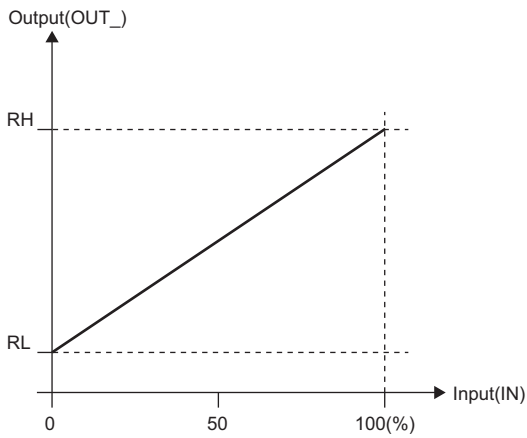
#### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
RH	Engineering Value High Limit	-999999 to 999999	100.0	User	REAL
RL	Engineering Value Low Limit	-999999 to 999999	0.0	User	REAL

## Processing details

### ■Engineering value conversion processing

This function block converts a value in percentage (%) input from the input variable IN into an engineering value, such as temperature and pressure and outputs the result from the output variable OUT\_.



$$OUT\_ = \left\{ (RH - RL) \times \frac{IN}{100} \right\} + RL$$

RH: Engineering value high limit

RL: Engineering value low limit

IN: Input value (0 to 100%)

OUT\_: Output value

This expression is also applicable for  $RH \leq RL$ .

## Operation error

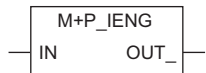
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (IN) or operation constant (RH, RL) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# 6.5 Engineering Value Inverse Conversion (M+P\_IENG)

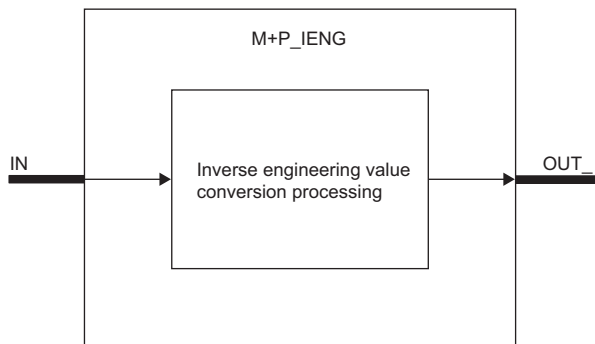
## M+P\_IENG

This FB converts an input engineering value (IN) such as temperature and pressure into a value in percentage (%) and outputs (OUT\_) the result.

### FBD/LD



### Block diagram



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN	Input	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL

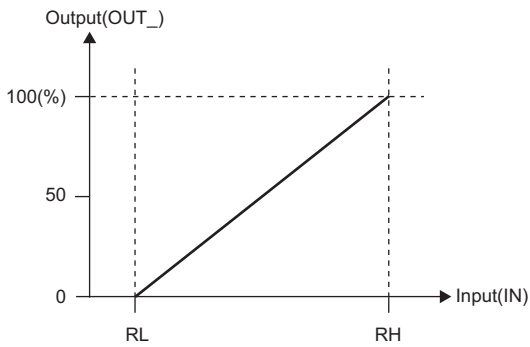
#### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
RH	Engineering Value High Limit	-999999 to 999999	100.0	User	REAL
RL	Engineering Value Low Limit	-999999 to 999999	0.0	User	REAL

## Processing details

### ■Engineering value inverse conversion processing

This function block converts an engineering value input from the input variable IN, such as temperature and pressure, into a value in percentage (%) and outputs the result from the output variable OUT\_.



$$\text{OUT}_-(\%) = \frac{\text{IN} - \text{RL}}{\text{RH} - \text{RL}} \times 100$$

RH: Engineering value high limit

RL: Engineering value low limit

IN: Input value

OUT\_: Output value (0 to 100%)

- This expression is also applicable for  $\text{RH} \leq \text{RL}$ .

## Operation error

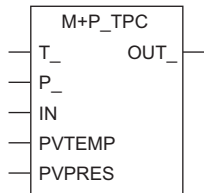
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (IN) or operation constant (RH, RL) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

## 6.6 Temperature/Pressure Correction (M+P\_TPC)

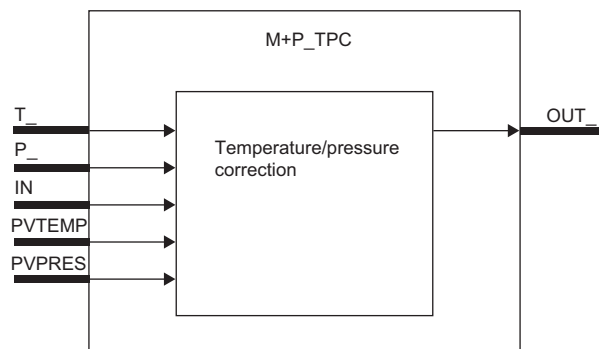
### M+P\_TPC

This FB executes temperature/pressure correction (or either of temperature correction or pressure correction) to the input (IN) of the differential pressure (%) and outputs (OUT\_) the result.

#### FBD/LD



#### Block diagram



#### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
T_	Temperature correction selection	TRUE: Used FALSE: Not used	Input variable	BOOL
P_	Pressure correction selection	TRUE: Used FALSE: Not used	Input variable	BOOL
IN	Differential pressure input	0 to 100[%]	Input variable	REAL
PVTEMP	Measured temperature (engineering value)	-999999 to 999999	Input variable	REAL
PVPRES	Measured pressure (engineering value)	-999999 to 999999	Input variable	REAL
OUT_	Output	0 to 100[%]	Output variable	REAL

#### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
TEMP	Design Temperature T' (engineering value)	-999999 to 999999	0.0	User	REAL
B1_	Bias Temperature (engineering value)	-999999 to 999999	273.15	User	REAL
PRES	Design Pressure P' (engineering value)	-999999 to 999999	0.0	User	REAL
B2_	Bias Pressure (engineering value)	-999999 to 999999	10332.0	User	REAL



## Processing details

### Temperature/pressure correction processing

This function block executes temperature/pressure correction (or either of temperature correction or pressure correction) to the differential pressure (%) value input from the input variable IN and outputs the result from the output variable OUT\_.

Temperature/pressure correction selection		A1	A2	Output (OUT_)
Temperature correction (T_)	Pressure correction (P_)			
TRUE	TRUE	$\frac{TEMP+B1\_}{PVTEMP+B1\_}$	$\frac{PVPRES+B2\_}{PRES+B2\_}$	OUT_ = IN × A1 × A2
FALSE	TRUE	1.0	$\frac{PVPRES+B2\_}{PRES+B2\_}$	
TRUE	FALSE	$\frac{TEMP+B1\_}{PVTEMP+B1\_}$	1.0	

T\_ : Temperature correction selection

P\_ : Pressure correction selection

IN: Differential pressure input (%)

PVTEMP: Measured temperature (engineering value)

PVPRES: Measured pressure (engineering value)

TEMP: Design temperature T' (engineering value)

PRES: Design pressure P' (engineering value)

B1\_ : Bias temperature (engineering value)

B2\_ : Bias pressure (engineering value)

OUT\_ : Output (%)

#### Point

Always input engineering values to the measured temperature (PVTEMP) and the measured pressure (PVPRES).

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (IN, PVTEMP, PVPRES) or operation constant (TEMP, B1_, PRES, B2_) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

## 6.7 Summation (M+P\_SUM)

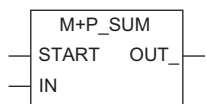
### M+P\_SUM

When the integration start signal (START) is TRUE, this function block performs integration processing to the input (IN) and outputs (OUT\_) the result.

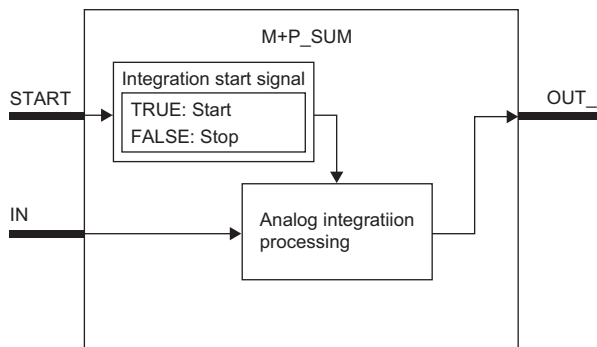
#### Point

Use M+P\_SUM2\_ to reduce the influence of information loss in the single-precision floating-point operation. (Page 98 M+P\_SUM2\_)  
M+P\_SUM is used to keep the compatibility with existing programs.

#### FBD/LD



#### Block diagram



#### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
START	Integration start signal	TRUE: Start FALSE: Stop	Input variable	BOOL
IN	Input	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL

#### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
ILC	Input Low Cut-off Value	-999999 to 999999	0.0	User	REAL
A_	Initial value	-999999 to 999999	0.0	User	REAL
RANGE	Input Range	1 to 3 1: /s 2: /min 3: /hour	1	User	INT

## Processing details

### ■Analog integration processing

When the integration start signal (START) is TRUE, this function block performs integration processing to the value input from the input variable IN and outputs the result from the output variable OUT\_.

Integration start signal (START)	Input (IN)	Output (OUT_)
FALSE (Stop)	Optional	OUT_ = Initial value (A_)
TRUE (Start)	IN ≤ ILC	OUT_ = Last value
	IN > ILC	OUT_ = $(IN \times \frac{\Delta T}{T}) + \text{previous value}$

$\Delta T$ : Execution cycle

ILC: Input low-cut value

A\_: Initial value

T: When RANGE is 1, T is 1 (s). When RANGE is 2, T is 60 (s). When RANGE is 3, T is 3600 (s).

#### Ex.

When a value of 0 to 5m<sup>3</sup>/min is input, set RANGE to 2 because the input range is "/min".

The multiplying factor is  $\times 1\text{m}^3$ .

### ■Integration start signal

When the integration start signal (START) is FALSE: The integration processing stops.

When the integration start signal (START) is TRUE: The integration processing starts.

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (START, IN), output data (OUT_), or operation constant (ILC, A_) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Input range (RANGE) is less than 1 or greater than 3.

#### Point

Use M+P\_SUM2\_ to reduce the influence of information loss in the single-precision floating-point operation. When M+P\_SUM is used, information loss error in the single-precision floating-point operation may occur in the integration operation (current value + up to the last integrated value).

Use M+P\_SUM to maintain the compatibility with existing programs.

"Information loss" is an error caused by round-down or round-up of a small value when the extremely small value is added to an extremely large value. This error occurs in a calculation system for the floating-point operation. The real numbers of the programmable controllers are expressed in single-precision floating point numbers.

The number of significant digits of real numbers in decimal is 6 or 7. An error occurs in the operation result of the following real number operation.

$$0.013333 + 32768.0 = 32768.013333 \rightarrow 32768.012$$

(current value + up to the last integrated value)

The logical operation result 32768.013333 is rounded down to 32768.012 and the increment 0.013333 decreases to 0.012. As the number of digits of the integral part of the integrated value increases, the number of significant digits of the decimal part decreases.

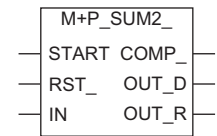
# 6.8 Summation (Internal Integer Integration) (M+P\_SUM2\_)

## M+P\_SUM2\_

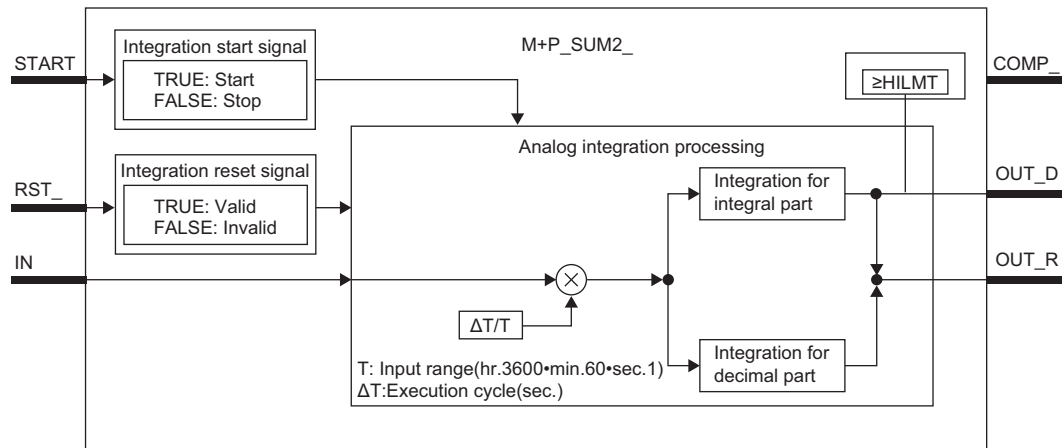
When the integration start signal (START) is TRUE, this function block performs integration processing to the input (IN) and outputs the result.

Internal integration for the integral part is executed with signed 32-bit integers.

### FBD/LD



### Block diagram



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
START	Integration start signal	TRUE: Start FALSE: Stop	Input variable	BOOL
RST_	Integration reset signal	TRUE: Valid FALSE: Invalid	Input variable	BOOL
IN	Input	-999999 to 999999	Input variable	REAL
COMP_	Integration complete signal	TRUE: Complete FALSE: Unreached	Output variable	BOOL
OUT_D	Integration value (integral part) output	-2147483648 to 2147483647	Output variable	DINT
OUT_R	Integration value real number output	-2147483648 to 2147483647	Output variable	REAL

#### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
ILC	Input Low Cut-off Value	-999999 to 999999	0.0	User	REAL
A_	Initial value	-999999 to 999999	0.0	User	REAL
RANGE	Input Range	1 to 3 1: /s 2: /min 3: /hour	1	User	INT
HILMT	High Limit Value of Integration	1 to 2147483647	1000000	User	DINT

Variable name	Description	Recommended range	Initial value	Set by	Data type
CYCLIC	TRUE: When the integrated value exceeds the high limit, the value returns to 0.*1 FALSE: When the integrated value exceeds the high limit, the high limit value is held.	TRUE, FALSE	TRUE	User	BOOL

\*1 The surplus to the integration high limit is added.  
For example, the integrated value will be 10 in the following condition: HILMT = 1000, Last integrated value = 990, Current value (IN × ΔT/T) = 20.

**Point**

For a REAL data type output value (output variable OUT\_R), the number of significant digits is 6 or 7 because this value is processed in 32-bit single-precision floating-point data. Consequently, a rounding error occurs when the integrated value exceeds the range of the number of significant digits, and the integral part may not match with the DINT type output value (output variable OUT\_D).

### Processing details

#### ■Analog integration processing

When the integration start signal (START) is TRUE and the integration reset signal (RST\_) is FALSE, this function block performs integration processing to the value input from the input variable IN and outputs the result from the output variable.

Integration start signal (START)	Integral reset signal (RST_)	Input (IN)	Output (OUT_D, OUT_R)
FALSE (Stop)	FALSE (Invalid)	Optional	OUT_D, OUT_R = Last value
	TRUE (Valid)	Optional	OUT_D, OUT_R = Initial value (A_)
TRUE (Start)	FALSE (Invalid)	IN ≤ ILC	OUT_D, OUT_R = Last value
		IN > ILC	OUT_D, OUT_R = (IN × $\frac{\Delta T}{T}$ ) + previous value
	TRUE (Valid)	Optional	OUT_D, OUT_R = Initial value (A_)

ΔT: Execution cycle  
ILC: Input low-cut value  
A\_: Initial value  
T: When RANGE is 1, T is 1 (s). When RANGE is 2, T is 60 (s). When RANGE is 3, T is 3600 (s).

**Ex.**

When a value of 0 to 5m<sup>3</sup>/min is input, set RANGE to 2 because the input range is "/min".  
The multiplying factor is ×1m<sup>3</sup>.

- The execution cycle (ΔT) indicates the execution cycle in the property setting of the FBD/LD program for the process control on the engineering tool, the cycle for starting up the timer in the program execution setting, and the interruption cycle of the fixed scan interrupt startup.

#### ■Integration start signal

When the integration start signal (START) is FALSE: The integration processing stops.  
When the integration start signal (START) is TRUE: The integration processing starts.

#### ■Integration reset signal

When the integration reset signal (RST\_) is FALSE: The integrated value is not reset (to the initial value).  
When the integration reset signal (RST\_) is TRUE: The integrated value is reset and the initial value is output.

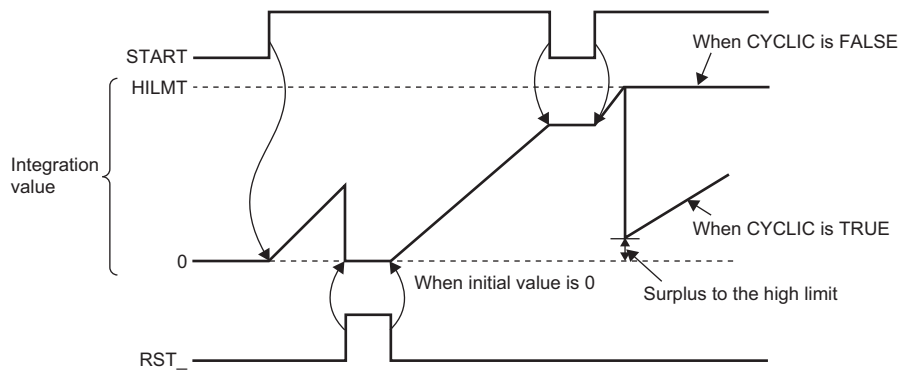
#### ■Integration complete signal

Integrated value output (integral part) (OUT\_D) ≥ Integration high limit (HILMT): The integration complete signal is TRUE\*1.  
Integrated value output (integral part) (OUT\_D) < Integration high limit (HILMT): The integration complete signal is FALSE.

\*1 When CYCLIC is TRUE, TRUE is output for only one cycle.

## ■ Timing chart

The following shows the timing chart for the integration start signal (START), integrated value, integration reset signal, and integration high limit.



## Operation error

Error code (SD0)	Description
3402H	Input data (IN, $\Delta T/T$ ) or operation constant (ILC, A_) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

## 6.9 Range Conversion (M+P\_RANGE\_)

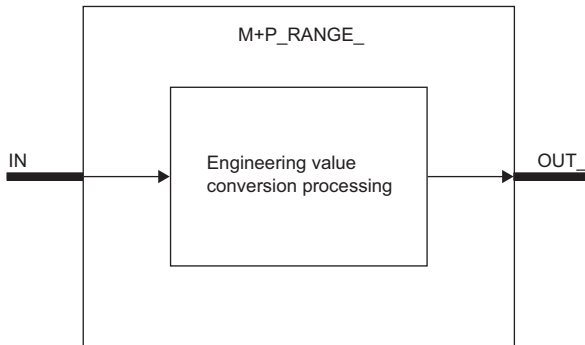
### M+P\_RANGE\_

This FB executes range conversion to the input (IN) and outputs (OUT\_) the result.

FBD/LD



#### ■Block diagram



6

#### Setting data

#### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN	Input	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL

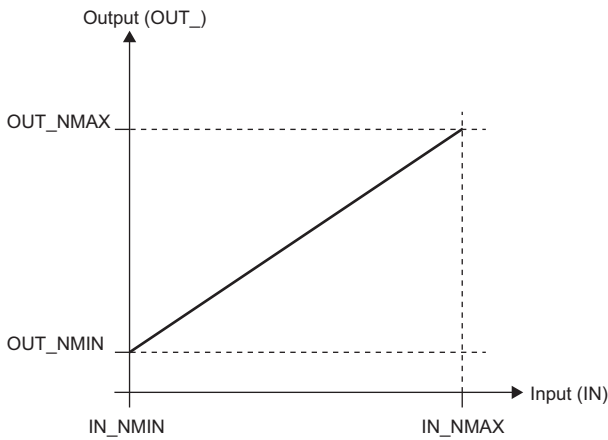
#### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
OUT_NMAX	Output High Limit Alarm	-999999 to 999999	100.0	User	REAL
OUT_NMIN	Output Low Limit Alarm	-999999 to 999999	0.0	User	REAL

## Processing details

### ■Engineering value conversion processing

This function block executes range conversion to the value input from the input variable IN and outputs the result from the output variable OUT\_.



$$OUT\_ = \frac{(OUT\_NMAX - OUT\_NMIN) \times (IN - IN\_NMIN)}{IN\_NMAX - IN\_NMIN} + OUT\_NMIN$$

OUT\_NMAX: Output High Limit Alarm

OUT\_NMIN: Output Low Limit Alarm

IN: Input value (IN\_NMIN to IN\_NMAX)

OUT\_: Output value

This expression is also applicable for  $OUT\_NMAX \leq OUT\_NMIN$  and  $IN\_NMAX \leq IN\_NMIN$ .

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (IN) or operation constant (IN_NMAX, IN_NMIN, OUT_NMAX, OUT_NMIN) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.



# 7 ARITHMETIC OPERATION


The following FBs perform operation processing such as addition/subtraction, multiplication/division, and square root.

## Point

The output variables and public variables of general process FBs are latched.

## Point

To set initial values of public variables of the general process FB arranged on a user-defined tag FB in "FB Property" of the engineering tool, refer to the following.

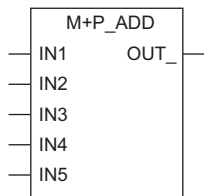
 GX Works3 Operating Manual

## 7.1 Addition (with coefficient) (M+P\_ADD)

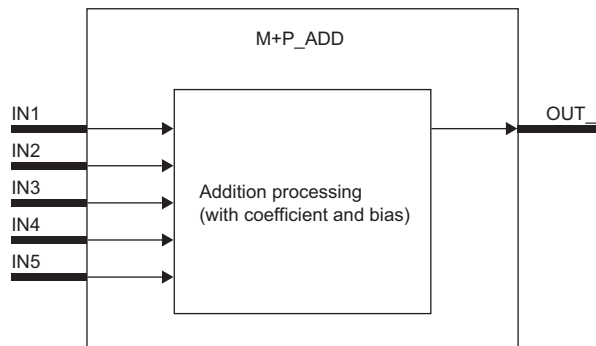
### M+P\_ADD

This FB adds values with coefficients and bias to the inputs (IN1 to IN5) and outputs (OUT\_) the result.

#### FBD/LD



#### Block diagram



#### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN1	Input 1	-999999 to 999999	Input variable	REAL
IN2	Input 2	-999999 to 999999	Input variable	REAL
IN3	Input 3	-999999 to 999999	Input variable	REAL
IN4	Input 4	-999999 to 999999	Input variable	REAL
IN5	Input 5	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
K_1 to K_5	Coefficient 1: Coefficient of IN1 data to Coefficient 5: Coefficient of IN5 data	-999999 to 999999	1.0	User	REAL
B_	Bias	-999999 to 999999	0.0	User	REAL

## Processing details

### Addition processing

This function block multiplies the values input from the input variables IN1 to IN5 with coefficients and bias and outputs the result from the output variable OUT\_.

Input (IN)	Output (OUT_)
IN1 to IN5	$OUT_ = (K_1 \times IN1) + (K_2 \times IN2) + (K_3 \times IN3) + (K_4 \times IN4) + (K_5 \times IN5) + B_$

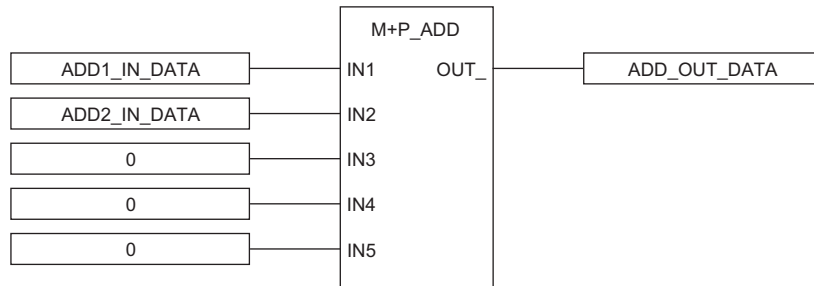
IN1 to IN5: Input value, K\_1 to K\_5: Coefficient, B\_: Bias

When no value to be input exists for any of the inputs (IN1 to IN5), input 0.

**Ex.**

When values are input only for IN1 and IN2

The value output from the output (OUT\_) will be calculated as follows:  $OUT_ = (K_1 \times IN1) + (K_2 \times IN2) + B_$ .



## Operation error

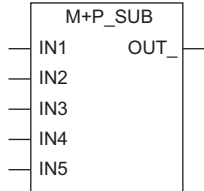
Error code (SD0)	Description
3402H	Input data (IN1 to IN5) or operation constant (K_1 to K_5 and B_) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

## 7.2 Subtraction (with coefficient) (M+P\_SUB)

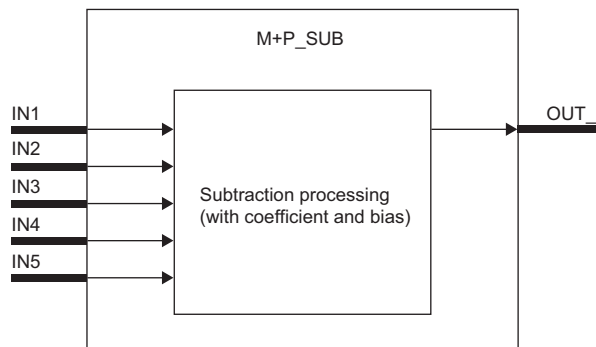
### M+P\_SUB

This FB subtracts values with coefficients and bias from the values of the inputs (IN1 to IN5) and outputs (OUT\_) the result.

#### FBD/LD



#### ■Block diagram



#### Setting data

#### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN1	Input 1	-999999 to 999999	Input variable	REAL
IN2	Input 2	-999999 to 999999	Input variable	REAL
IN3	Input 3	-999999 to 999999	Input variable	REAL
IN4	Input 4	-999999 to 999999	Input variable	REAL
IN5	Input 5	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL

#### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
K_1 to K_5	Coefficient 1: Coefficient of IN1 data to Coefficient 5: Coefficient of IN5 data	-999999 to 999999	1.0	User	REAL
B_	Bias	-999999 to 999999	0.0	User	REAL

## Processing details

### ■ Subtraction processing

This function block subtracts values with coefficients and bias from the values input from the input variables IN1 to IN5 and outputs the result from the output variable OUT\_.

Input (IN)	Output (OUT_)
IN1 to IN5	$OUT_ = (K_1 \times IN1) - (K_2 \times IN2) - (K_3 \times IN3) - (K_4 \times IN4) - (K_5 \times IN5) + B_$

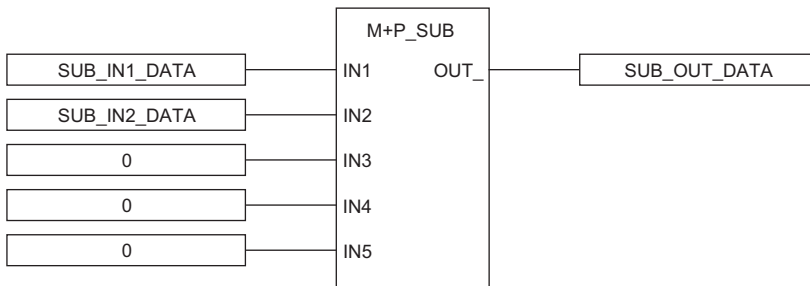
IN1 to IN5: Input, K\_1 to K\_5: Coefficient, B\_ : Bias

When no value to be input exists for any of the inputs (IN1 to IN5), input 0.

**Ex.**

When values are input only for IN1 and IN2

The value output from the output (OUT\_) will be calculated as follows:  $OUT_ = (K_1 \times IN1) - (K_2 \times IN2) + B_$ .



## Operation error

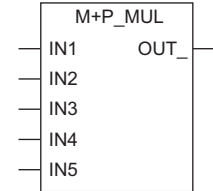
Error code (SD0)	Description
3402H	Input data (IN1 to IN5) or operation constant (K_1 to K_5 and B_) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# 7.3 Multiplication (with coefficient) (M+P\_MUL)

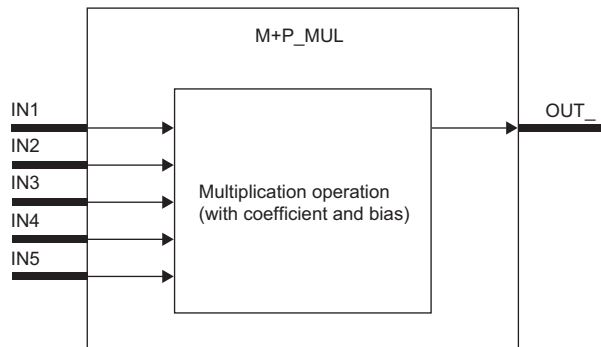
## M+P\_MUL

This FB multiplies the values of the inputs (IN1 to IN5) with coefficients and bias and outputs (OUT\_) the result.

### FBD/LD



### Block diagram



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN1	Input 1	-999999 to 999999	Input variable	REAL
IN2	Input 2	-999999 to 999999	Input variable	REAL
IN3	Input 3	-999999 to 999999	Input variable	REAL
IN4	Input 4	-999999 to 999999	Input variable	REAL
IN5	Input 5	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL

#### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
K_1 to K_5	Coefficient 1: Coefficient of IN1 data to Coefficient 5: Coefficient of IN5 data	-999999 to 999999	1.0	User	REAL
B_	Bias	-999999 to 999999	0.0	User	REAL

## Processing details

### ■ Multiplication processing

This function block multiplies the values input from the input variables IN1 to IN5 with coefficients and bias and outputs the result from the output variable OUT\_.

Input (IN)	Output (OUT_)
IN1 to IN5	$OUT_ = (K_1 \times IN1) \times (K_2 \times IN2) \times (K_3 \times IN3) \times (K_4 \times IN4) \times (K_5 \times IN5) + B_$

IN1 to IN5: Input, K\_1 to K\_5: Coefficient, B\_ : Bias

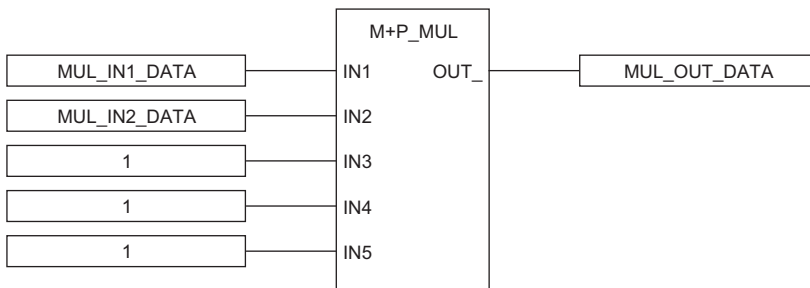
#### Point

When no value to be input exists for any of the inputs (IN1 to IN5), input 1 to both the input and coefficient.  
When either input or coefficient is set to 0, the bias (B\_) will be output (OUT\_).

#### Ex.

When values are input only for IN1 and IN2

The value output from the output (OUT\_) will be calculated as follows:  $OUT_ = (K_1 \times IN1) \times (K_2 \times IN2) + B_$ .



## Operation error

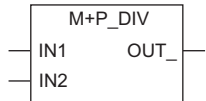
Error code (SD0)	Description
3402H	Input data (IN1 to IN5) or operation constant (K_1 to K_5 and B_) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

## 7.4 Division (with coefficient) (M+P\_DIV)

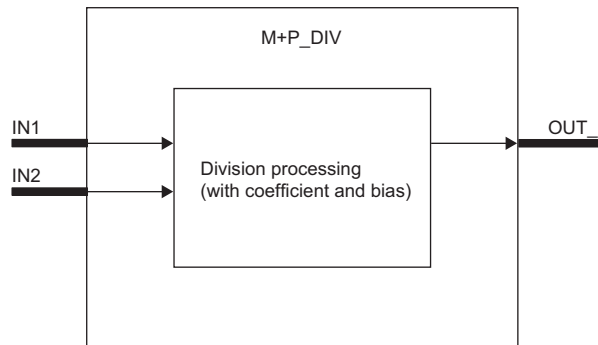
### M+P\_DIV

This FB divides the values of the inputs (IN1, IN2) with coefficients and bias and outputs (OUT\_) the result.

FBD/LD



### ■Block diagram



### Setting data

#### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN1	Input 1	-999999 to 999999	Input variable	REAL
IN2	Input 2	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL

#### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
A_	Coefficient	-999999 to 999999	1.0	User	REAL
K_1	Coefficient 1: Coefficient of IN1 data	-999999 to 999999	1.0	User	REAL
K_2	Coefficient 2: Coefficient of IN2 data	-999999 to 999999	1.0	User	REAL
B1_	IN1 data bias	-999999 to 999999	0.0	User	REAL
B2_	IN2 data bias	-999999 to 999999	0.0	User	REAL
B3_	Bias	-999999 to 999999	0.0	User	REAL

## Processing details

### ■ Division processing

This function block divides the values input from the input variables IN1 to IN2 with coefficients and bias and outputs the result from the output variable OUT\_.

Input (IN2), coefficient (K_2), bias (B2_): Denominator	Output (OUT_)
$K_2 \times IN2 + B2_$ is not 0 (denominator $\neq 0$ )	$OUT_ = A_ \times \frac{K_1 \times IN1 + B1_}{K_2 \times IN2 + B2_} + B3_$
$K_2 \times IN2 + B2_$ is 0 (denominator = 0)	$OUT_ = B3_$

IN1 to IN2: Input value, A\_ and K\_1 to K\_2: Coefficient, B1\_ to B3\_: Bias

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (IN1 to IN2) or operation constant (A_, K_1, K_2, B1_, B2_, and B3_) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

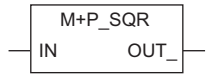


# 7.5 Square Root (with coefficient) (M+P\_SQR)

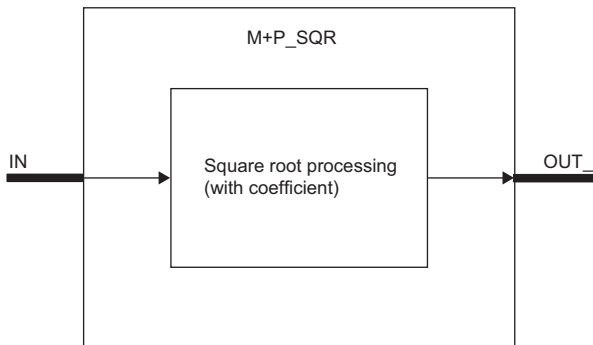
## M+P\_SQR

This FB executes square root extraction with coefficients to the value of the input (IN) and outputs (OUT\_) the result.

### FBD/LD



### Block diagram



7

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN	Input	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL

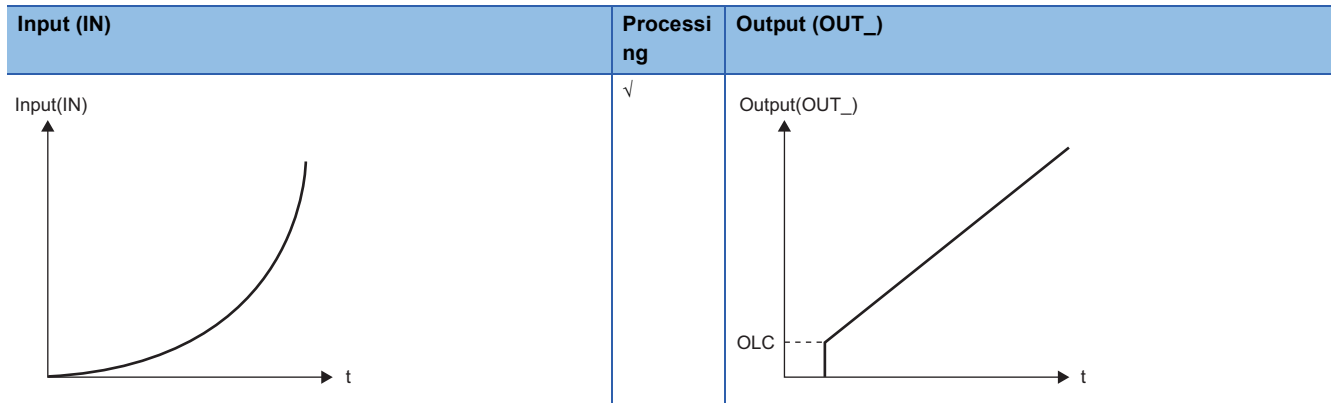
#### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
OLC	Output Low Cut-off Value	0 to 999999	0.0	User	REAL
K_	Coefficient	0 to 999999	10.0	User	REAL

## Processing details

### ■ Square root processing

This function block executes square root extraction with coefficients to the value input from the input variable IN and outputs the result from the output variable OUT\_.

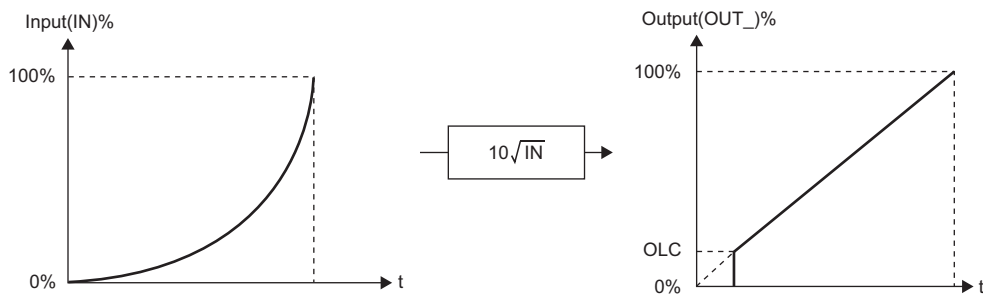


Input (IN)	Output low cut-off value (OLC)	Output (OUT_)
IN ≥ 0	$K_ \times \sqrt{IN} > OLC$	$OUT_ = K_ \times \sqrt{IN}$
	$K_ \times \sqrt{IN} \leq OLC$	$OUT_ = 0$
IN < 0	—	$OUT_ = 0$

IN: Input value, K\_: Coefficient

### Point

- When the value of the input (IN) is in percentage (%), set the coefficient (K\_) to 10. When square root extraction is executed with  $K_ = 10$  ( $10\sqrt{IN}$ ), 0 to 100% of the input (IN) correspond to 0 to 100% of the output (OUT\_). (For input = 100%, output =  $10\sqrt{100} = 100\%$ )



- Generally, the output low cut-off value (OLC) is used when the value of the input (IN) is in percentage (%). (The output low cut-off value (OLC) must be set to around 10 (%) level.)

## Operation error

Error code (SD0)	Description
3402H	Input data (IN) or operation constant (OLC, K_) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# 8 COMPARISON OPERATION


The following FBs perform comparison operation ( $\geq$ ,  $>$ ,  $=$ ,  $<$ ,  $\leq$ ).

## Point

The output variables and public variables of general process FBs are latched.

## Point

To set initial values of public variables of the general process FB arranged on a user-defined tag FB in "FB Property" of the engineering tool, refer to the following.

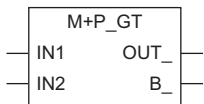
 GX Works3 Operating Manual

## 8.1 Comparison (>) with setting value (M+P\_GT)

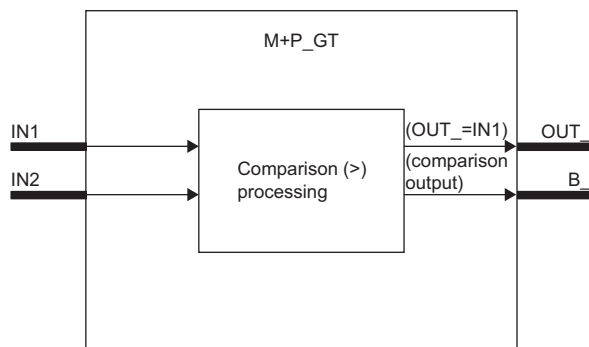
### M+P\_GT

This FB compares (>) the input 1 (IN1) with the input 2 (IN2) using a setting value and hysteresis and outputs the result to the comparison output (B\_). This function block always outputs values to the input 1 (IN1) from the output (OUT\_).

#### FBD/LD



#### Block diagram



#### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN1	Input 1	-999999 to 999999	Input variable	REAL
IN2	Input 2	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL
B_	Comparison Output	TRUE, FALSE	Output variable	BOOL

#### Public variable (operation constant)

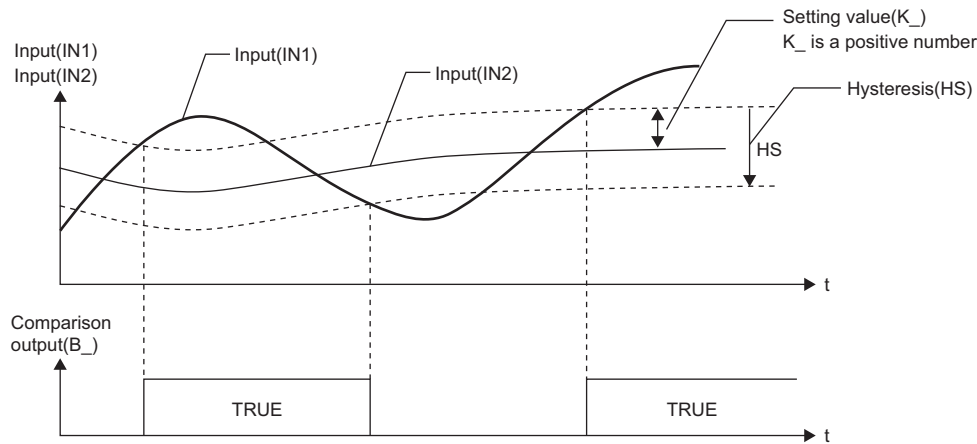
Variable name	Description	Recommended range	Initial value	Set by	Data type
K_	Setting Value	-999999 to 999999	0.0	User	REAL
HS	Hysteresis	0 to 999999	0.0	User	REAL

## Processing details

### ■ Comparison (>) processing

This function block compares (>) the value input from the input variable IN1 with the value input from the input variable IN2 using a setting value and hysteresis and outputs the comparison result from the output variable B\_.

This function block always outputs values to the input variable IN1 from the output variable OUT\_.



Condition	Comparison output (B_)	Output (OUT_)
$IN1 > IN2 + K_$	B_ = TRUE	OUT_ = IN1
$IN1 \leq IN2 + K_ - HS$	B_ = FALSE	
$IN2 + K_ - HS < IN1 \leq IN2 + K_$	Last value	

IN1: Input value 1

IN2: Input value 2

K\_: Setting value

HS: Hysteresis

## Operation error

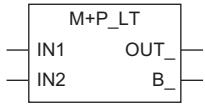
Error code (SD0)	Description
3402H	Input data (IN1, IN2) or operation constant (K_, HS) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Hysteresis (HS) is less than 0.

## 8.2 Comparison (<) with setting value (M+P\_LT)

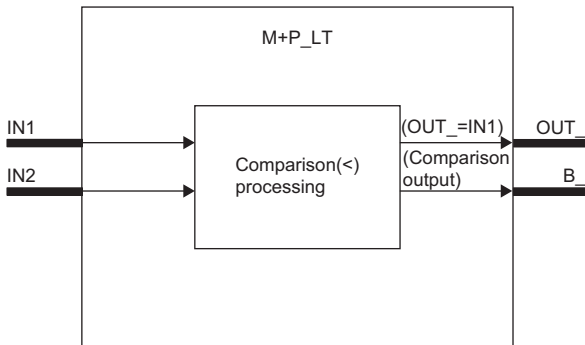
### M+P\_LT

This FB compares (<) the input 1 (IN1) with the input 2 (IN2) using a setting value and hysteresis and outputs the result to the comparison output (B\_). This function block always outputs values to the input 1 (IN1) from the output (OUT\_).

#### FBD/LD



### ■Block diagram



### Setting data

#### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN1	Input 1	-999999 to 999999	Input variable	REAL
IN2	Input 2	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL
B_	Comparison Output	TRUE, FALSE	Output variable	BOOL

#### ■Public variable (operation constant)

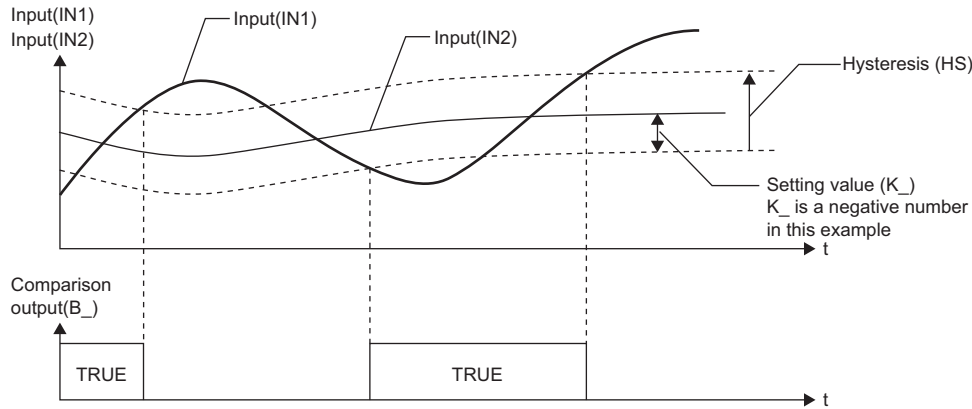
Variable name	Description	Recommended range	Initial value	Set by	Data type
K_	Setting Value	-999999 to 999999	0.0	User	REAL
HS	Hysteresis	0 to 999999	0.0	User	REAL

## Processing details

### ■ Comparison (<) processing

This function block compares (<) the value input from the input variable IN1 with the value input from the input variable IN2 using a setting value and hysteresis and outputs the comparison result from the output variable B\_.

This function block always outputs values to the input variable IN1 from the output variable OUT\_.



Condition	Comparison output (B_)	Output (OUT_)
$IN1 < IN2 + K_-$	$B_- = \text{TRUE}$	$OUT_- = IN1$
$IN1 \geq IN2 + K_- + HS$	$B_- = \text{FALSE}$	
$IN2 + K_- \leq IN1 < IN2 + K_- + HS$	Last value	

IN1: Input value 1  
 IN2: Input value 2  
 K\_: Setting value  
 HS: Hysteresis

## Operation error

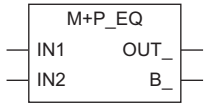
Error code (SD0)	Description
3402H	Input data (IN1, IN2) or operation constant (K_, HS) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Hysteresis (HS) is less than 0.

## 8.3 Comparison (=) with setting value (M+P\_EQ)

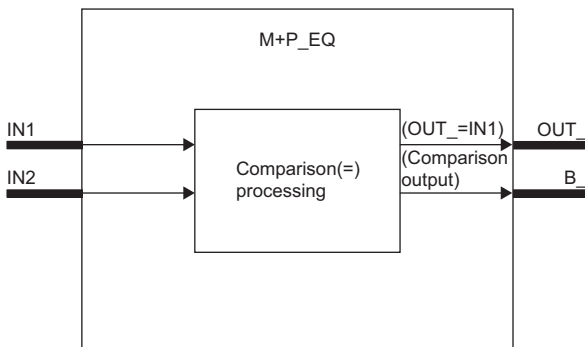
### M+P\_EQ

This FB compares (=) the input 1 (IN1) with the input 2 (IN2) using a setting value and outputs the result to the comparison output (B\_). This function block always outputs values to the input 1 (IN1) from the output (OUT\_).

#### FBD/LD



### ■Block diagram



### Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN1	Input 1	-999999 to 999999	Input variable	REAL
IN2	Input 2	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL
B_	Comparison Output	TRUE, FALSE	Output variable	BOOL

### ■Public variable (operation constant)

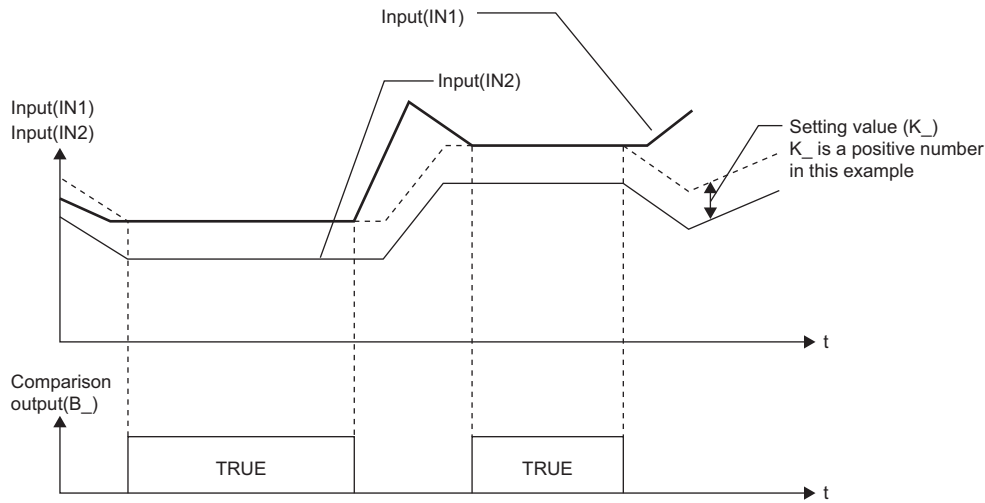
Variable name	Description	Recommended range	Initial value	Set by	Data type
K_	Setting Value	-999999 to 999999	0.0	User	REAL

## Processing details

### ■ Comparison (=) processing

This function block compares (=) the value input from the input variable IN1 with the value input from the input variable IN2 using a setting value and outputs the comparison result from the output variable B\_.

This function block always outputs values to the input variable IN1 from the output variable OUT\_.



Condition	Comparison output (B_)	Output (OUT_)
$IN1 = IN2 + K_$	B_ = TRUE	OUT_ = IN1
$IN1 \neq IN2 + K_$	B_ = FALSE	

IN1: Input value 1, IN2: Input value 2, K\_: Setting value

## Operation error

Error code (SD0)	Description
3402H	Input data (IN1, IN2) or operation constant (K_) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

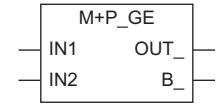


# 8.4 Comparison ( $\geq$ ) with setting value (M+P\_GE)

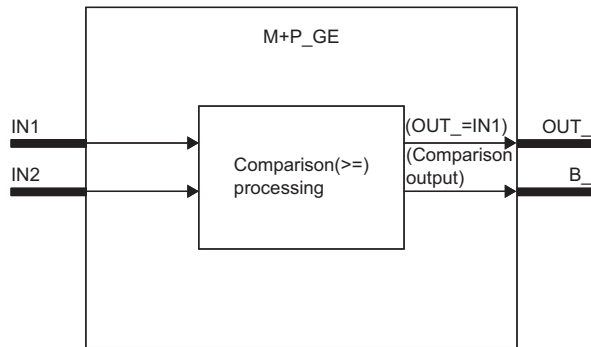
## M+P\_GE

This FB compares ( $\geq$ ) the input 1 (IN1) with the input 2 (IN2) using a setting value and hysteresis and outputs the result to the comparison output (B\_). This function block always outputs values to the input 1 (IN1) from the output (OUT\_).

### FBD/LD



### Block diagram



### Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN1	Input 1	-999999 to 999999	Input variable	REAL
IN2	Input 2	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL
B_	Comparison Output	TRUE, FALSE	Output variable	BOOL

### Public variable (operation constant)

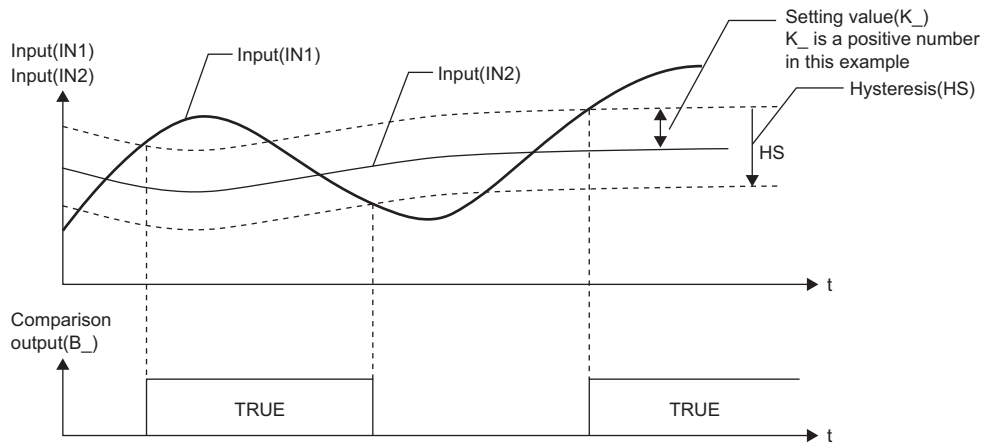
Variable name	Description	Recommended range	Initial value	Set by	Data type
K_	Setting Value	-999999 to 999999	0.0	User	REAL
HS	Hysteresis	0 to 999999	0.0	User	REAL

## Processing details

### ■ Comparison ( $\geq$ ) processing

This function block compares ( $\geq$ ) the value input from the input variable IN1 with the value input from the input variable IN2 using a setting value and hysteresis and outputs the comparison result from the output variable B\_.

This function block always outputs values to the input variable IN1 from the output variable OUT\_.



Condition	Comparison output (B_)	Output (OUT_)
$IN1 \geq IN2 + K_$	B_ = TRUE	OUT_ = IN1
$IN1 < IN2 + K_ - HS$	B_ = FALSE	
$IN2 + K_ - HS \leq IN1 < IN2 + K_$	Last value	

IN1: Input value 1

IN2: Input value 2

K\_ : Setting value

HS: Hysteresis

## Operation error

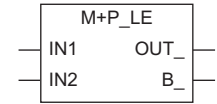
Error code (SD0)	Description
3402H	Input data (IN1, IN2) or operation constant (K_, HS) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Hysteresis (HS) is less than 0.

# 8.5 Comparison (<=) with setting value (M+P\_LE)

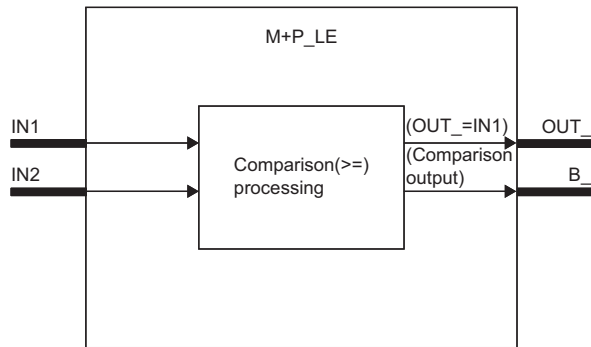
## M+P\_LE

This FB compares ( $\leq$ ) the input 1 (IN1) with the input 2 (IN2) using a setting value and hysteresis and outputs the result to the comparison output (B\_). This function block always outputs values to the input 1 (IN1) from the output (OUT\_).

### FBD/LD



### Block diagram



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN1	Input 1	-999999 to 999999	Input variable	REAL
IN2	Input 2	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL
B_	Comparison Output	TRUE, FALSE	Output variable	BOOL

#### Public variable (operation constant)

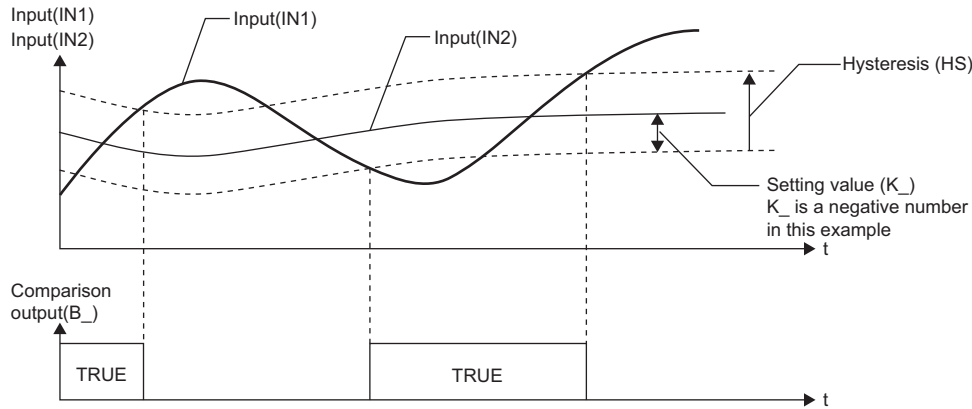
Variable name	Description	Recommended range	Initial value	Set by	Data type
K_	Setting Value	-999999 to 999999	0.0	User	REAL
HS	Hysteresis	0 to 999999	0.0	User	REAL

## Processing details

### ■ Comparison ( $\leq$ ) processing

This function block compares ( $\leq$ ) the value input from the input variable IN1 with the value input from the input variable IN2 using a setting value and hysteresis and outputs the comparison result from the output variable B\_.

This function block always outputs values to the input variable IN1 from the output variable OUT\_.



Condition	Comparison output (B_)	Output (OUT_)
$IN1 \leq IN2 + K_$	B_ = TRUE	OUT_ = IN1
$IN1 > IN2 + K_ + HS$	B_ = FALSE	
$IN2 + K_ < IN1 \leq IN2 + K_ + HS$	Last value	

IN1: Input value 1  
 IN2: Input value 2  
 K\_: Setting value  
 HS: Hysteresis

## Operation error

Error code (SD0)	Description
3402H	Input data (IN1, IN2) or operation constant (K_, HS) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Hysteresis (HS) is less than 0.

# 9 CONTROL OPERATION

The following FBs control operation of lead-lag, integration, derivative, high/low limiter, variation rate limiter, dead band, bumpless transfer, and analog memory.

**Point** The output variables and public variables of general process FBs are latched.

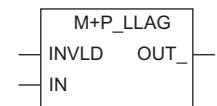
**Point** To set initial values of public variables of the general process FB arranged on a user-defined tag FB in "FB Property" of the engineering tool, refer to the following.  
 GX Works3 Operating Manual

## 9.1 Lead-Lag (M+P\_LLAG)

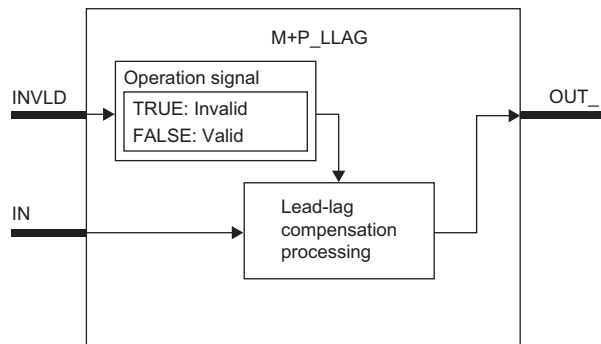
### M+P\_LLAG

When the operation signal (INVLD) is FALSE, this function block performs lead-lag compensation to the input (IN) and outputs (OUT\_) the result.

#### FBD/LD



#### Block diagram



#### Setting data

##### Input/output variable

Variable name	Description	Recommended range	Type	Data type
INVLD	Operation signal	TRUE: Invalid FALSE: Valid	Input variable	BOOL
IN	Input	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL

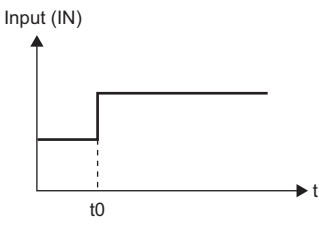
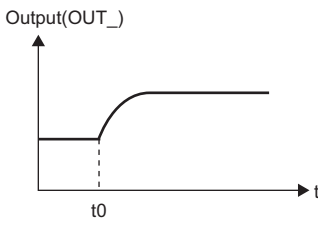
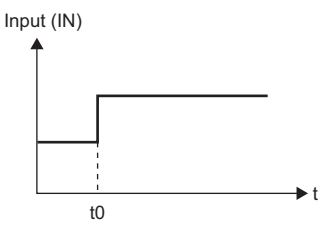
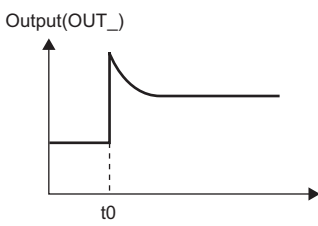
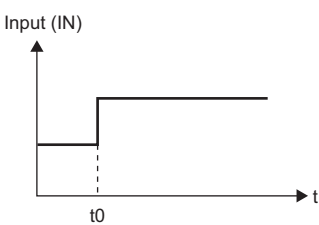
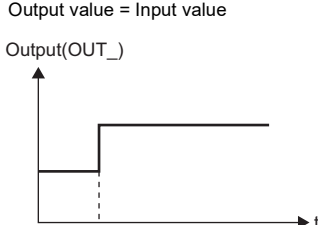
##### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
T1_	Lag time (lag time constant)	0 to 999999[s]	1.0	User	REAL
T2_	Lead time (lead time constant)	0 to 999999[s]	1.0	User	REAL

## Processing details

### Lead-lag compensation processing

When the operation signal (INVLD) is FALSE, this function block performs lead-lag compensation to the value input from the input variable IN and outputs the result from the output variable OUT\_.

Condition		Input (IN)	Lead-lag compensation	Output (OUT_)
Operation signal (INVLD)	Lag time (T1_) Lead time (T2_)			
FALSE	T1_ > T2_		$\frac{1+T2_ \cdot S}{1+T1_ \cdot S}$	
	T1_ < T2_		$\frac{1+T2_ \cdot S}{1+T1_ \cdot S}$	
TRUE	T1_ > T2_ T1_ < T2_		None	Output value = Input value 

Operation signal (INVLD)	Output (OUT_)
FALSE (Valid)	$OUT_ = \frac{1}{T1_ + \Delta T} \{ (T2_ \times (IN - IN_{n-1})) + (T1_ \times OUT_{n-1}) + (\Delta T \times IN) \}$ <ul style="list-style-type: none"> <li>• For T1_ + ΔT = 0, OUT_ = 0</li> <li>• When the lead time (lead time constant) T2_ is 0, the output is the same as the output of when the primary lag filter is applied.</li> </ul>
TRUE (Invalid)	OUT_ = IN

IN: Input value  
 OUT\_: Output value  
 IN<sub>n-1</sub>: Last input value  
 OUT<sub>n-1</sub>: Last output value  
 T1\_: Lag time (lag time constant) (s)  
 T2\_: Lead time (lead time constant) (s)  
 ΔT: Execution cycle (s)  
 S: Laplace operator

### Operation signal

When the operation signal (INVLD) is FALSE: Lead-lag compensation is valid.

When the operation signal (INVLD) is TRUE: Lead-lag compensation is invalid.

## Operation error

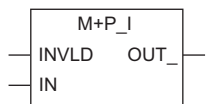
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (INVLD, IN), output data (OUT_), or operation constant (T1_, T2_) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The lag time (T1) is less than 0 or lead time (T2) is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.

## 9.2 Integration (M+P\_I)

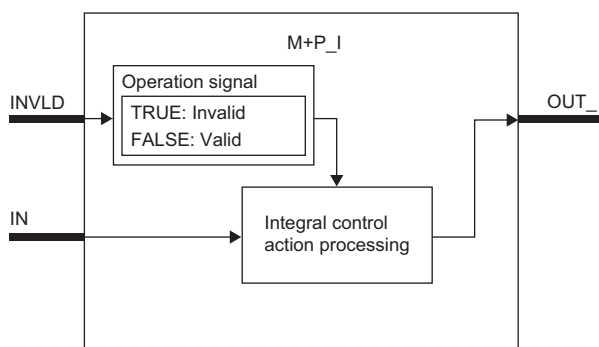
### M+P\_I

When the operation signal (INVLD) is FALSE, this function block performs integral operation to the input (IN) and outputs (OUT\_) the result.

#### FBD/LD



### ■Block diagram



### Setting data

#### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
INVLD	Operation signal	TRUE: Invalid FALSE: Valid	Input variable	BOOL
IN	Input	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL

#### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
T_	Integral Time	0 to 999999[s]	1.0	User	REAL
Ys	Initial Output Value	-999999 to 999999	0.0	User	REAL



## Processing details

### ■ Integral operation processing

When the operation signal (INVLD) is FALSE, this function block performs integral operation to the value input from the input variable IN and outputs the result from the output variable OUT\_.

Condition		Input (IN)	Integral operation	Output (OUT_)
Operation signal (INVLD)	Integral time (T_)			
FALSE	T_ > 0		$\frac{1}{T_ \cdot S}$	
	T_ = 0	<p>When the integral time (T_) after t2 is 0</p>	$\frac{1}{T_ \cdot S}$	
TRUE	T_ ≥ 0		Ys	

Operation signal (INVLD)	Integral time (T_)	Output (OUT_)
FALSE (Valid)	T_ > 0	$OUT_ = \frac{\Delta T \times IN}{T_} + OUT_{n-1}$
	T_ = 0	$OUT_ = OUT_{n-1}$
TRUE (Invalid)	—	$OUT_ = Ys$

- ΔT: Execution cycle
- T\_: Integral time (s)
- IN: Input value
- OUT\_: Output value
- OUT<sub>n-1</sub>: Last output value
- Ys: Initial output value
- S: Laplace operator

### ■ Operation signal

When the operation signal (INVLD) is FALSE: Integral operation is valid.

When the operation signal (INVLD) is TRUE: Integral operation is invalid.

## Operation error

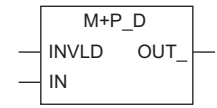
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (INVLD, IN), output data, or operation constant (T_, Ys) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# 9.3 Derivative (M+P\_D)

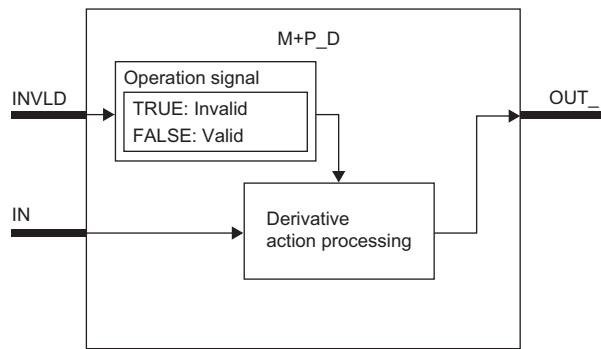
## M+P\_D

When the operation signal (INVLD) is FALSE, this function block performs derivative operation to the input (IN) and outputs (OUT\_) the result.

### FBD/LD



### Block diagram



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
INVLD	Operation signal	TRUE: Invalid FALSE: Valid	Input variable	BOOL
IN	Input	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL

#### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
T_	Derivative Time	0 to 999999[s]	1.0	User	REAL
Ys	Initial Output Value	-999999 to 999999	0.0	User	REAL

## Processing details

### Derivative operation processing

When the operation signal (INVLD) is FALSE, this function block performs derivative operation to the value input from the input variable IN and outputs the result from the output variable OUT\_.

Condition	Input (IN)	Derivative operation	Output (OUT_)
Operation signal (INVLD) FALSE		$\frac{T_-\cdot S}{1+T_-\cdot S}$	
TRUE		Ys	

Operation signal (INVLD)	Output (OUT_)
FALSE (Valid)	$OUT_ = \frac{T_-}{T_- + \Delta T} \times (OUT_{n-1} - IN_{n-1} + IN)$
TRUE (Invalid)	OUT_ = Ys

$\Delta T$ : Execution cycle  
 $T_-$ : Derivative time (s)  
 IN: Input value  
 $IN_{n-1}$ : Last input value  
 $OUT_-$ : Output value  
 $OUT_{n-1}$ : Last output value  
 Ys: Initial output value  
 S: Laplace operator

### Operation signal

When the operation signal (INVLD) is FALSE: Derivative operation is valid.

When the operation signal (INVLD) is TRUE: Derivative operation is invalid.

## Operation error

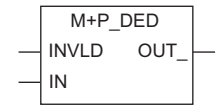
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (INVLD, IN), output data, or operation constant ( $T_-$ , Ys) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# 9.4 Dead Time (M+P\_DED)

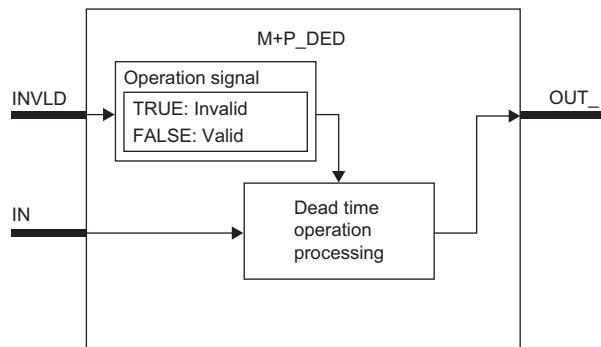
## M+P\_DED

When the operation signal (INVLD) is FALSE, this function block outputs (OUT\_) the result with a delay by the dead time to the input (IN).

### FBD/LD



### Block diagram



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
INVLD	Operation signal	TRUE: Invalid FALSE: Valid	Input variable	BOOL
IN	Input	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL

#### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
ST_	Data Collection Interval	0 to 9999[s]	1.0	User	REAL
SN	Sampling Count	0 to 48	0	User	INT
Ys	Initial Output Value	-999999 to 999999	0.0	User	REAL
OCHG	Output Switching When Initialized	0, 1	0	User	INT

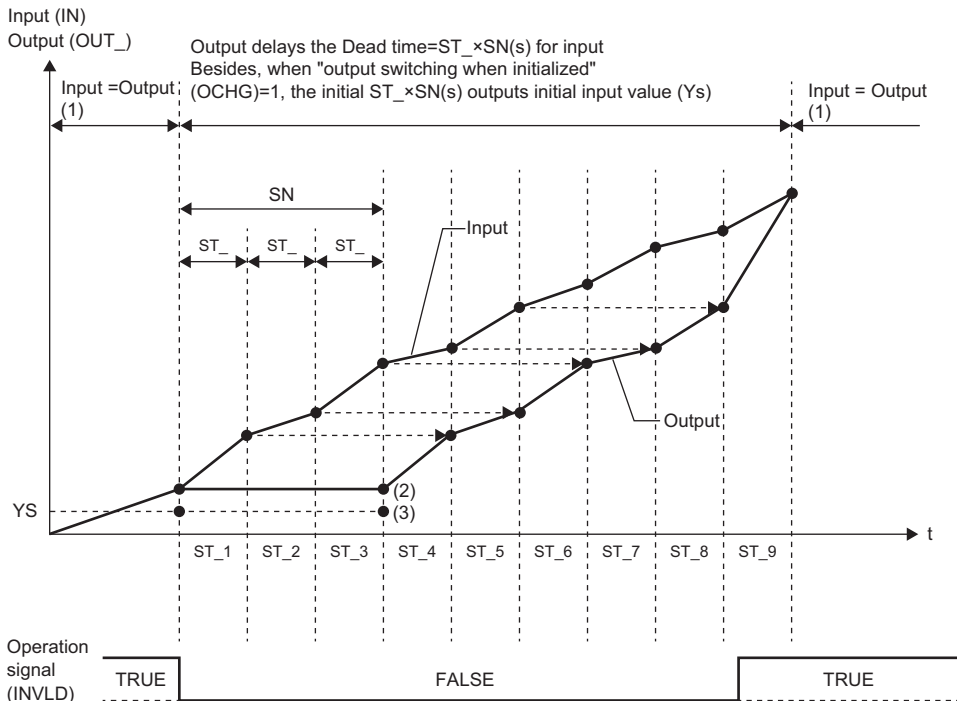
# Processing details

## Dead time processing

When the operation signal (INVLD) is FALSE, this function block outputs the input value from the input variable IN with a delay by the dead time from the output variable OUT\_.

Ex.

When the sampling count (SN) is 3



No.	Operation signal (INVLD)	Output switching when initialized (OCHG)	Dead time	Output (OUT_)
(1)	TRUE (Invalid)	Optional (0 or 1)	—	Outputs (OUT_) the input (IN) value.
(2)	FALSE (Valid)	0	$ST_ \times SN$	Outputs (OUT_) the input values (IN) that are input when the operation signal (INVLD) changes from TRUE to FALSE from the beginning of operation to the SN time. After the SN time, OUT_ outputs the value of $IN_n - SN$ .
(3)		1	$ST_ \times SN$	Outputs (OUT_) the initial output value (Ys) from the beginning of operation to the SN time. After the SN time, OUT_ outputs the value of $IN_n - SN$ .

INVLD: Operation signal

OCHG: Output switching when initialized

Ys: Initial output value

IN: Input value

OUT\_: Output value

ST\_: Data collection interval (s)

SN: Sampling count

- When the sampling count (SN) is 0, the output (OUT\_) value equals to the input (IN) value.

## Operation signal

When the operation signal (INVLD) is FALSE: Dead time operation is valid.

When the operation signal (INVLD) is TRUE: Dead time operation is invalid.

## Operation error

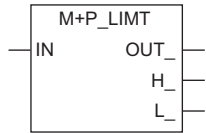
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (INVLD, IN) or operation constant (ST_, SN, Ys, OCHG) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# 9.5 High/Low Limiter (M+P\_LIMT)

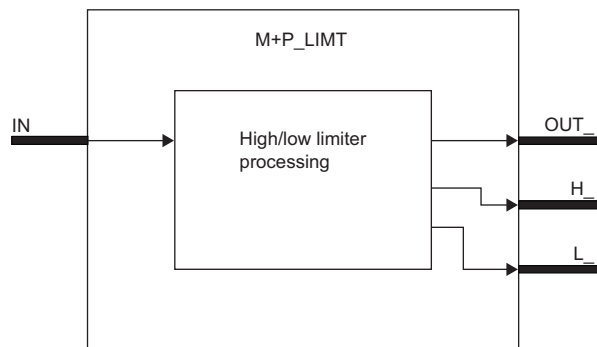
## M+P\_LIMT

This FB applies high/low limiters with hysteresis to the input (IN) and outputs (OUT\_) the result.

### FBD/LD



### Block diagram



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN	Input	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL
H_	High limit over detection	TRUE: Detected FALSE: Reset	Output variable	BOOL
L_	Low limit over detection	TRUE: Detected FALSE: Reset	Output variable	BOOL

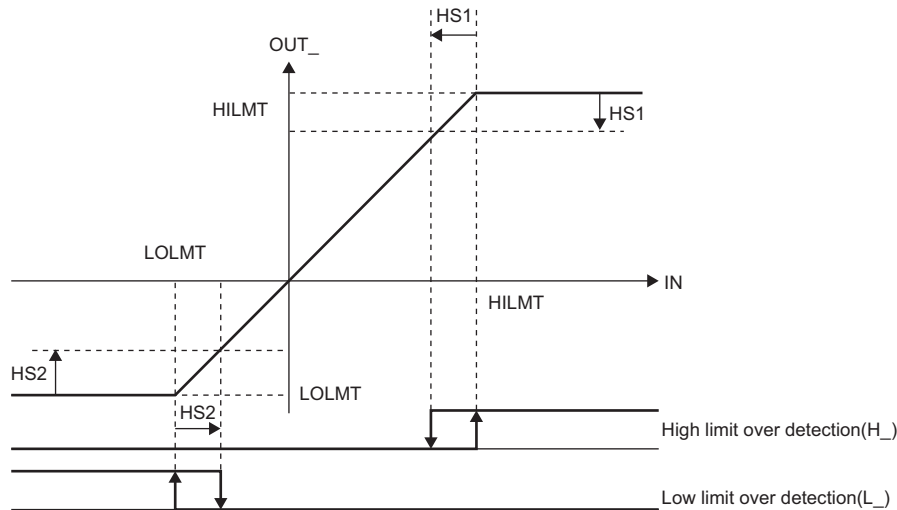
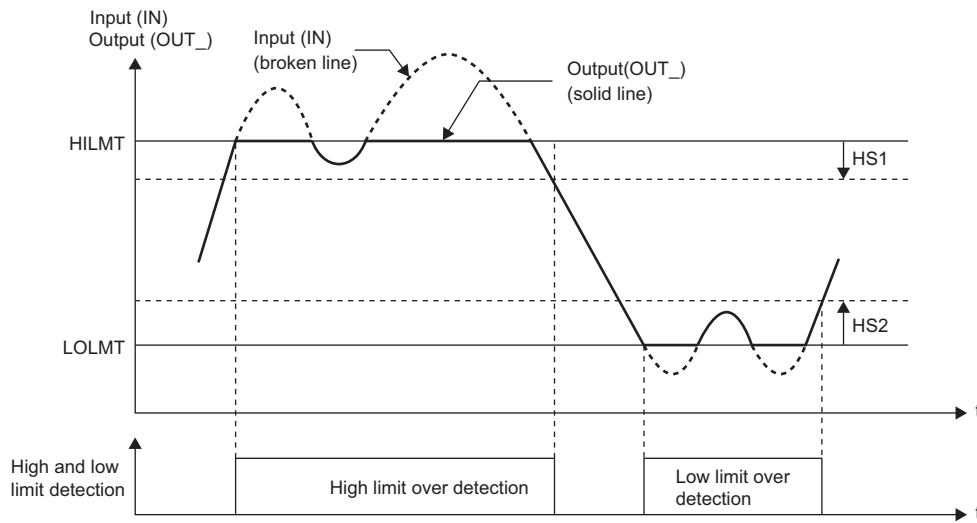
#### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
HILMT	High Limit Value	-999999 to 999999	100.0	User	REAL
LOLMT	Low Limit Value	-999999 to 999999	0.0	User	REAL
HS1	High Limit Hysteresis	0 to 999999	0.0	User	REAL
HS2	Low Limit Hysteresis	0 to 999999	0.0	User	REAL



**High/low limiter processing**

This function block applies high/low limiters with hysteresis to the value input from the input variable IN and outputs the result from the output variable OUT\_.



IN: Input value  
 OUT\_: Output value  
 HILMT: High limit value  
 LOLMT: Low limit value  
 HS1: High limit hysteresis  
 HS2: Low limit hysteresis

Input (IN)	Output (OUT_)	High limit over detection (H_)	Low limit over detection (L_)
$IN \geq HILMT$	HILMT	TRUE (Detected)	FALSE (Canceled)
$LOLMT + HS2 < IN < HILMT - HS1$	IN	FALSE (Canceled)	FALSE (Canceled)
$IN \leq LOLMT$	LOLMT	FALSE (Canceled)	TRUE (Detected)
Others	IN	Last value	Last value

Set the high limit value (HILMT) and low limit value (LOLMT) in such a way that HILMT equals to or exceeds LOLMT. When HS1 is less than 0 or HS2 is less than 0, an operation error occurs.

Condition	Result
$HS1 \geq 0$ and $HS2 \geq 0$	Normal
$HS1 \geq 0$ and $HS2 < 0$	Operation error
$HS1 < 0$ and $HS2 \geq 0$	
$HS1 < 0$ and $HS2 < 0$	

## Operation error

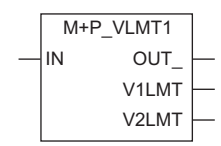
Error code (SD0)	Description
3402H	Input data (IN) or operation constant (HILMT, LOLMT, HS1, or HS2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The high limit hysteresis (HS1) is less than 0 or low limit hysteresis (HS2) is less than 0.
	The low limit value (LOLMT) is greater than high limit value (HILMT).

# 9.6 Variation Rate Limiter 1 (M+P\_VLMT1)

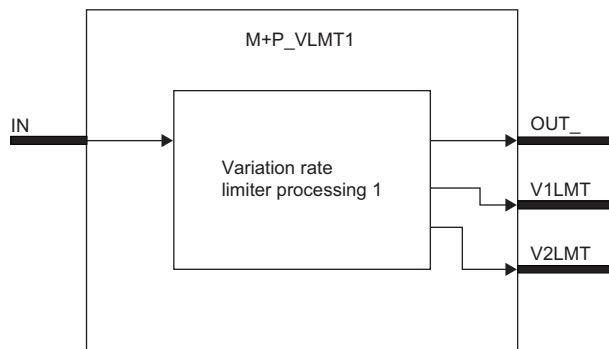
## M+P\_VLMT1

This FB limits the variation speed to the input (IN) and outputs (OUT\_) the result.

### FBD/LD



### Block diagram



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN	Input	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL
V1LMT	Positive direction limit	TRUE: Limited FALSE: Canceled	Output variable	BOOL
V2LMT	Negative direction limit	TRUE: Limited FALSE: Canceled	Output variable	BOOL

#### Public variable (operation constant)

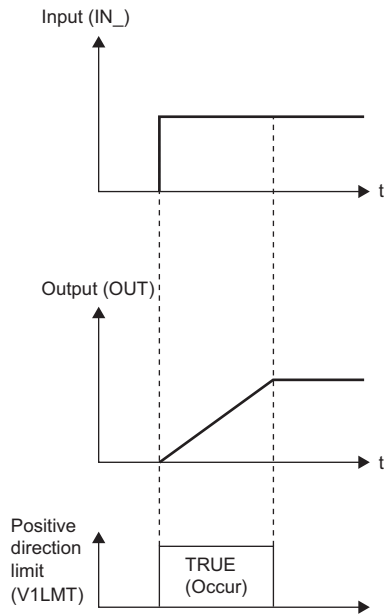
Variable name	Description	Recommended range	Initial value	Set by	Data type
V1_	Positive direction limit value	0 to 999999[1/s]	100.0	User	REAL
V2_	Negative direction limit value	0 to 999999[1/s]	100.0	User	REAL
HS1	Positive Direction Hysteresis	0 to 999999	0.0	User	REAL
HS2	Negative Direction Hysteresis	0 to 999999	0.0	User	REAL

## Processing details

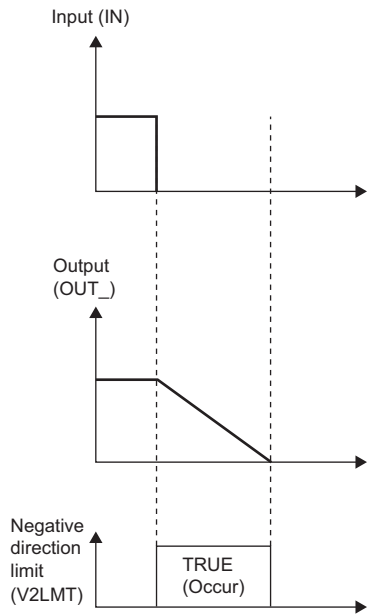
### ■ Variation rate limiter 1 processing

This function block limits the variation speed to the value input from the input variable IN and outputs the result from the output variable OUT\_.

• Positive direction



• Negative direction



• Positive direction  $IN \geq OUT_$

Input (IN-OUT_)	Output (OUT_)	Positive direction limit (V1LMT)	Negative direction limit (V2LMT)
$IN - OUT_ \geq V1_ \times \Delta T$	$OUT_ = OUT_ + V1_ \times \Delta T$	TRUE (Detected)	FALSE (Canceled)
$IN - OUT_ < V1_ \times \Delta T - HS1$	$OUT_ = IN$	FALSE (Canceled)	FALSE (Canceled)
Others	$OUT_ = IN$	Last value	Last value

• Negative direction  $IN < OUT_$

Input (IN-OUT_)	Output (OUT_)	Positive direction limit (V1LMT)	Negative direction limit (V2LMT)
$OUT_ - IN \geq V2_ \times \Delta T$	$OUT_ = OUT_ - V2_ \times \Delta T$	FALSE (Canceled)	TRUE (Detected)
$OUT_ - IN < V2_ \times \Delta T - HS2$	$OUT_ = IN$	FALSE (Canceled)	FALSE (Canceled)
Others	$OUT_ = IN$	Last value	Last value

$\Delta T$ : Execution cycle

IN: Input value

OUT\_: Output value

V1\_: Positive direction limit value (/s)

V2\_: Negative direction limit value (/s)

HS1: Positive direction hysteresis

HS2: Negative direction hysteresis

When HS1 is less than 0 or HS2 is less than 0, an operation error occurs.

Condition	Result
$HS1 \geq 0$ and $HS2 \geq 0$	Normal
$HS1 \geq 0$ and $HS2 < 0$	Operation error
$HS1 < 0$ and $HS2 \geq 0$	Operation error
$HS1 < 0$ and $HS2 < 0$	Operation error

## Operation error

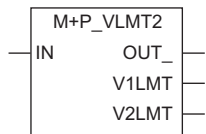
Error code (SD0)	Description
3402H	Input data (IN) or operation constant (V1_, V2_, HS1, HS2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The positive direction hysteresis (HS1) is less than 0 or negative direction hysteresis (HS2) is less than 0.

# 9.7 Variation Rate Limiter 2 (M+P\_VLMT2)

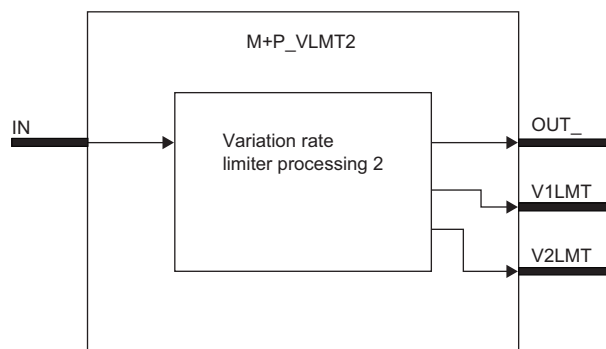
## M+P\_VLMT2

This FB limits the output variation speed to the input (IN) and outputs (OUT\_) the result.

### FBD/LD



### ■Block diagram



### Setting data

#### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN	Input	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL
V1LMT	Positive direction limit	TRUE: Limited FALSE: Canceled	Output variable	BOOL
V2LMT	Negative direction limit	TRUE: Limited FALSE: Canceled	Output variable	BOOL

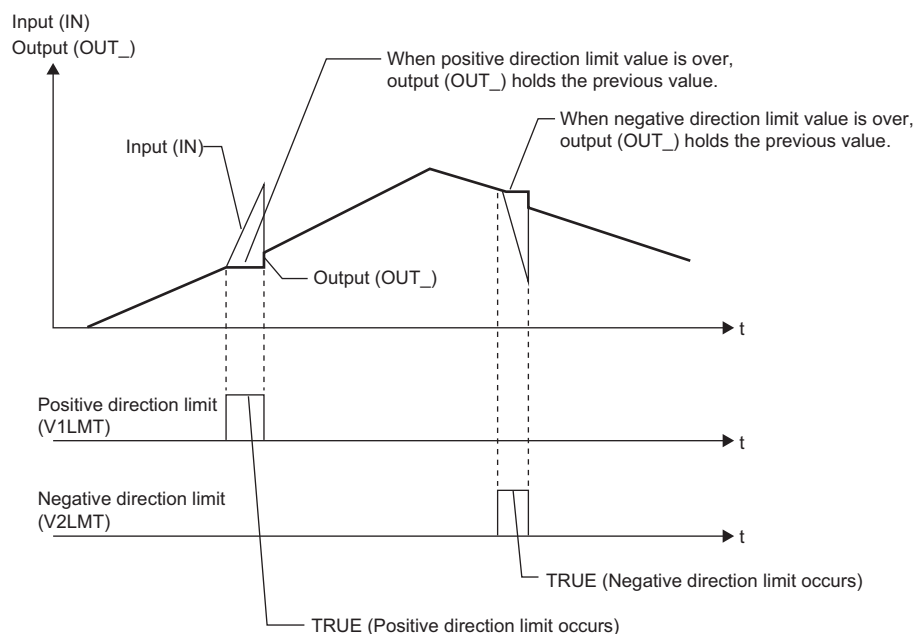
#### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
V1_	Positive direction limit value	0 to 999999	100.0	User	REAL
V2_	Negative direction limit value	0 to 999999	100.0	User	REAL
HS1	Positive Direction Hysteresis	0 to 999999	0.0	User	REAL
HS2	Negative Direction Hysteresis	0 to 999999	0.0	User	REAL

## Processing details

### ■ Variation rate limiter 2 processing

This function block limits the variation speed to the value input from the input variable IN and outputs the result from the output variable OUT\_.



When the variation rate of the input (IN) is greater than the limit value, the value of the output (OUT\_) is limited as shown above.

The last value is held if the variation rate limit value is exceeded.

Holding the last value is canceled when the variation rate falls below the limit value.

- Positive direction  $IN \geq OUT_$

Input (IN-OUT_)	Output (OUT_)	Positive direction limit (V1LMT)	Negative direction limit (V2LMT)
$IN - OUT_ \geq V1\_ \times \Delta T$	$OUT_ = OUT_$	TRUE (Detected)	FALSE (Canceled)
$IN - OUT_ < V1\_ \times \Delta T - HS1$	$OUT_ = IN$	FALSE (Canceled)	FALSE (Canceled)
Others	$OUT_ = OUT_$	Last value	Last value

- Negative direction  $IN < OUT_$

Input (IN-OUT_)	Output (OUT_)	Positive direction limit (V1LMT)	Negative direction limit (V2LMT)
$OUT_ - IN \geq V2\_ \times \Delta T$	$OUT_ = OUT_$	FALSE (Canceled)	TRUE (Detected)
$OUT_ - IN < V2\_ \times \Delta T - HS2$	$OUT_ = IN$	FALSE (Canceled)	FALSE (Canceled)
Others	$OUT_ = OUT_$	Last value	Last value

$\Delta T$ : Execution cycle

IN: Input value

OUT\_: Output value

V1\_: Positive direction limit value (/s)

V2\_: Negative direction limit value (/s)

HS1: Positive direction hysteresis

HS2: Negative direction hysteresis

When HS1 is less than 0 or HS2 is less than 0, an operation error occurs.

Condition	Result
$HS1 \geq 0$ and $HS2 \geq 0$	Normal
$HS1 \geq 0$ and $HS2 < 0$	Operation error
$HS1 < 0$ and $HS2 \geq 0$	Operation error
$HS1 < 0$ and $HS2 < 0$	Operation error

## Operation error

Error code (SD0)	Description
3402H	Input data (IN) or operation constant (V1_, V2_, HS1, HS2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The positive direction hysteresis (HS1) is less than 0 or negative direction hysteresis (HS2) is less than 0.

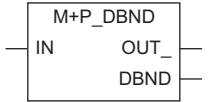


# 9.8 Dead Band (M+P\_DBND)

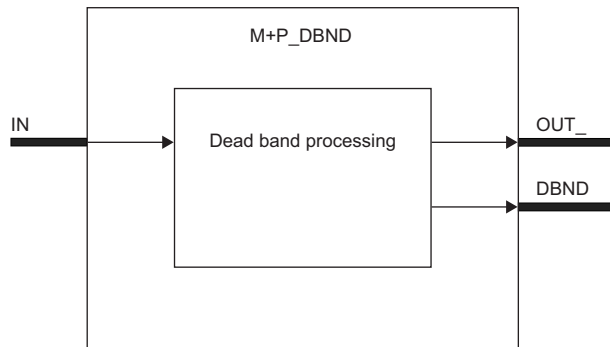
## M+P\_DBND

This FB sets a dead band to the input (IN) and outputs (OUT\_) the result.

### FBD/LD



### Block diagram



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN	Input	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL
DBND	Dead band range	TRUE: Within range FALSE: Out of range	Output variable	BOOL

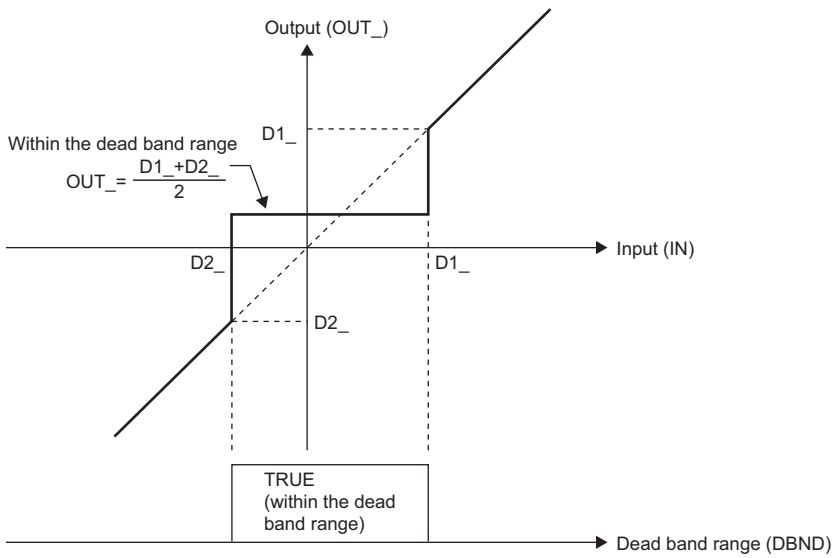
#### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
D1_	Dead Zone High Limit	-999999 to 999999	0.0	User	REAL
D2_	Dead Zone Low Limit	-999999 to 999999	0.0	User	REAL

## Processing details

### ■ Dead band processing

This function block sets a dead band for the value input from the input variable IN and outputs the result from the output variable OUT\_.



Input (IN)	Output (OUT_)	Dead band range (DBND)
$D2_ \leq IN \leq D1_$	$\frac{D1_ + D2_}{2}$	TRUE (within range)
$IN < D2_ \text{ or } IN > D1_$	IN	FALSE (out of range)

D1\_: Dead zone high limit  
D2\_: Dead zone low limit  
IN: Input value  
OUT\_: Output value  
DBND: Dead band range

## Operation error

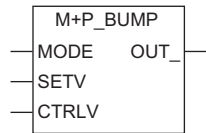
Error code (SD0)	Description
3402H	Input data (IN) or operation constant (D1_, D2_) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

## 9.9 Bump-less Transfer (M+P\_BUMP)

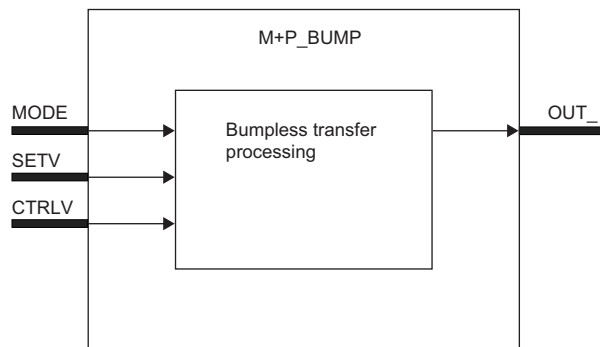
### M+P\_BUMP

This FB changes the output (OUT\_) from the output control value CTRLV to the output setting value SETV smoothly when the mode (MODE) is changed from FALSE (MANUAL) to TRUE (AUTO).

#### FBD/LD



#### ■Block diagram



#### Setting data

#### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
MODE	Mode switching	TRUE: AUTO FALSE: MANUAL	Input variable	BOOL
SETV	Output setting value	-999999 to 999999	Input variable	REAL
CTRLV	Output control value	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL

#### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
T_	Lag time	0 to 999999[s]	1.0	User	REAL
a_	Delay Band	0 to 999999	1.0	User	REAL

## Processing details

### ■ Bumpless transfer processing

This function block changes the values output from the output variable  $OUT\_$  from the output control value  $CTRLV$  to the output setting value  $SETV$  smoothly when the input variable  $MODE$  (mode switching) changes from  $FALSE$  (MANUAL) to  $TRUE$  (AUTO).

The output ( $OUT\_$ ) approaches the output setting value ( $SETV$ ) at a ratio set with the lag time ( $T\_$ ).

However, when the value is within the range set by the lag band ( $a\_$ ) based on the output setting value ( $SETV$ ), the output ( $OUT$ ) approaches the output setting value ( $SETV$ ) with the primary delay filter enabled.

Condition		$Xp$	Output ( $OUT\_ $ )
Mode switching (MODE)	$ Xp $		
FALSE (MANUAL)	—	$Xq = CTRLV - SETV$ $Xp = CTRLV - SETV$	$OUT\_ = CTRLV$
TRUE (AUTO)	$ Xp  > a\_ $	$Xp = Xp' - \frac{\Delta T}{T\_ } \times Xq$	$OUT\_ = SETV + Xp$ This expression is applicable when the following conditions are satisfied. $OUT\_ = SETV, Xp = Xp'$ $ Xp  \leq (\frac{\Delta T}{T\_ }) \times  Xq $
	$ Xp  \leq a\_ $	$Xp = \frac{T\_ }{T\_ + \Delta T} \times Xp'$	$OUT\_ = SETV + Xp$ This expression is applicable when the following conditions are satisfied. $OUT\_ = SETV, Xp = Xp'$ $ Xp  \leq 0.0001$

MODE: Mode switching  
 $OUT\_$ : Output value  
 $SETV$ : Output setting value  
 $CTRLV$ : Output control value  
 $Xq$ : Initial deviation  
 $Xp$ : Deviation  
 $\Delta T$ : Execution cycle  
 $T\_$ : Lag time (s)  
 $a\_$ : Delay band

## Operation error

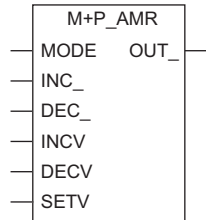
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (MODE, SETV, or CTRLV), operation constant ( $T\_ $ , $a\_ $ ), or internal operation value ( $Xq$ , $Xp$ ) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

## 9.10 Analog Memory (M+P\_AMR)

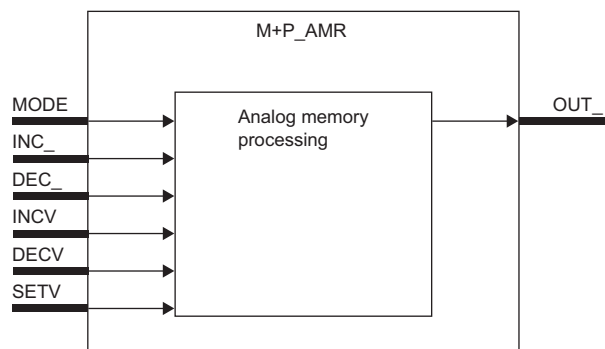
### M+P\_AMR

This FB increases or decreases the output (OUT\_) at a fixed rate.

#### FBD/LD



#### Block diagram



#### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
MODE	Mode switching	TRUE: AUTO FALSE: MANUAL	Input variable	BOOL
INC_	Output addition	TRUE: Used FALSE: Not used	Input variable	BOOL
DEC_	Output subtraction	TRUE: Used FALSE: Not used	Input variable	BOOL
INCV	Output addition value	-999999 to 999999	Input variable	REAL
DECV	Output subtraction value	-999999 to 999999	Input variable	REAL
SETV	Output setting value	-999999 to 999999	Input variable	REAL
OUT_	Output	-999999 to 999999	Output variable	REAL

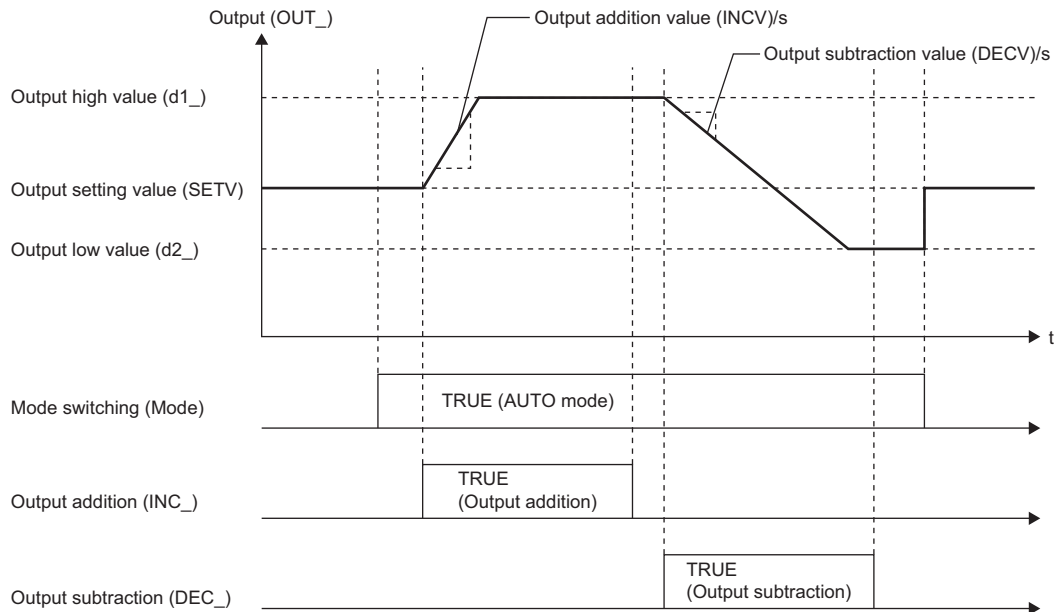
#### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
d1_	Output High Limit Value	0 to 999999	1.0	User	REAL
d2_	Output Low Limit Value	0 to 999999	1.0	User	REAL

## Processing details

### ■ Analog memory processing

This function block increases or decreases values at a fixed rate and outputs the result from the output variable OUT\_.



Condition			Output (OUT_)
Mode switching (MODE)	Output addition (INC_)	Output subtraction (DEC_)	
FALSE (MANUAL)	—	—	$OUT_ = SETV$
TRUE (AUTO)	TRUE	FALSE	$OUT_ = OUT_ +  INCV  \times \Delta T$ When $OUT_$ is more than $d1_$ , $OUT_$ equals to $d1_$ .
	FALSE	TRUE	$OUT_ = OUT_ -  DECV  \times \Delta T$ When $OUT_$ is less than $d2_$ , $OUT_$ equals to $d2_$ .
	TRUE	TRUE	$OUT_ = OUT_$
	FALSE	FALSE	$OUT_ = OUT_$

MODE: Mode switching  
 OUT\_: Output value  
 INC\_: Output addition signal  
 DEC\_: Output subtraction signal  
 INCV: Output addition value  
 DECV: Output subtraction value  
 SETV: Output setting value  
 d1\_: Output high limit value  
 d2\_: Output low limit value  
 ΔT: Execution cycle

## Operation error

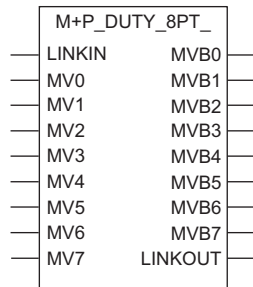
Error code (SD0)	Description
3402H	Input data (MODE, INC_, DEC_, INCV, DECV, or SETV) or operation constant (d1_, d2_) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

## 9.11 8 Points Time Proportional Output (M+P\_DUTY\_8PT\_)

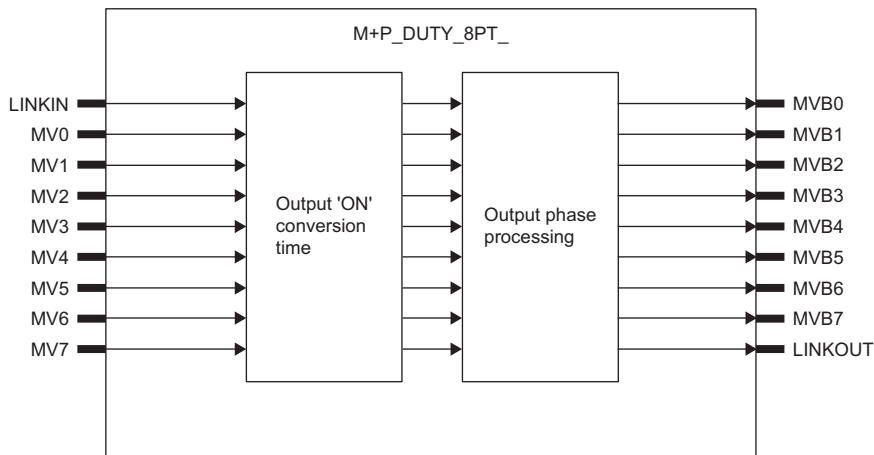
### M+P\_DUTY\_8PT\_

This FB performs output ON time conversion for input values and outputs the result in bits. This FB also adjusts the phase of output cycles automatically to suppress overlapping of output (for example, peak current).

#### FBD/LD



#### Block diagram



#### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
LINKIN	Link input (Indirect address)	—	Input variable	DWORD
MV0 to MV7	MV input	-10 to 110[%]	Input variable	REAL
MVB0 to MVB7	Bit ON/OFF duty output	TRUE, FALSE	Output variable	BOOL
LINKOUT	Link output (Indirect address)	—	Output variable	DWORD

#### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
PRIMARY	Lead FB specified	TRUE, FALSE	TRUE	User	BOOL
CTDUTY	Control Output Cycle	0 to 9999[s]	1.0	User	REAL
DELAY	Output ON Delay Time	0 to 9999[s]	0.0	User	REAL

## Processing details

### ■ Output ON time conversion/Output phase processing

This function block captures an input value (MVn) every control output cycle (CTDUTY) and outputs a duty manipulated value (MVBn) to the input value.

At the same time, this function block adjusts the phase of output cycles automatically to suppress peak current.

Item	Description
Duty manipulated value (MVBn) ON time	Duty manipulated value (MVBn) ON time = ON time execution cycle count × ΔT The ON time execution cycle count is defined as the following calculation result with the first digit after the decimal point rounded off. $\left( \frac{CTDUTY \times MVn}{\Delta T \times 100} \right)$
Duty manipulated value (MVBn) OFF time	Duty manipulated value (MVBn) OFF time = OFF time execution cycle count × ΔT The OFF time execution cycle count is defined as (number of executions in a control output cycle) - (execution cycle count of ON time).

CTDUTY: Control output cycle

ΔT: Execution cycle

MVn: Input to the nth pin (%)

MVBn: Output from the nth pin (BOOL)

- Interval of each bit ON/OFF duty output

Output of each pin starts with a delay for the time set with DELAY after the previous pin output turns off.

When the input value from an input pin is 0%, the delay time for the output pin is ignored.

#### Ex.

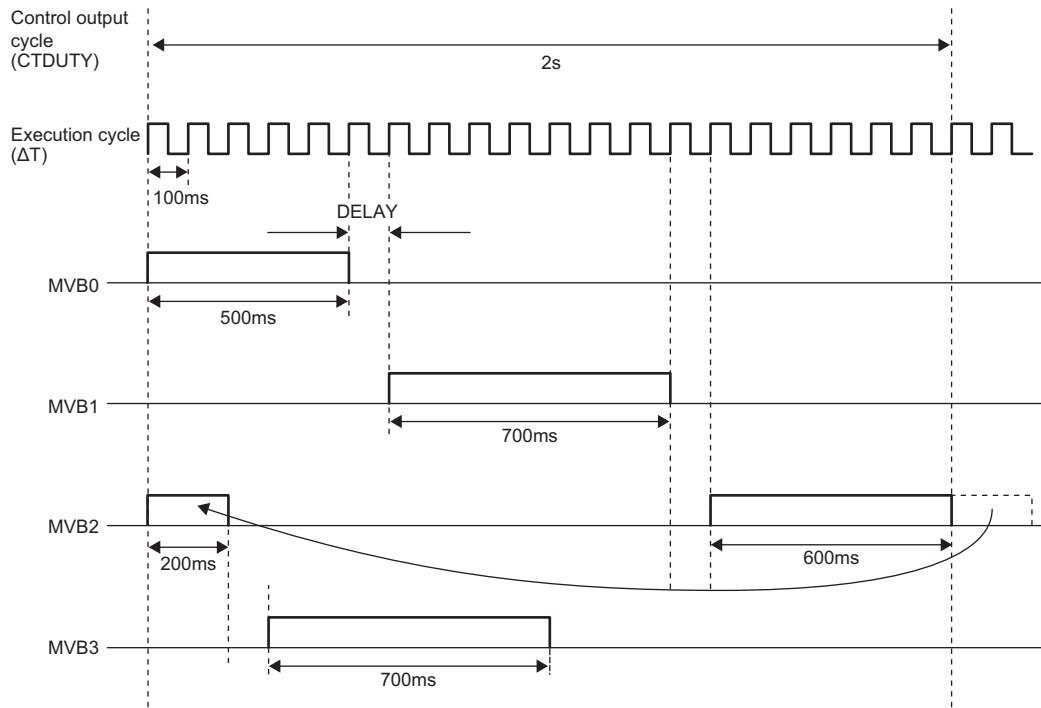
Execution cycle: ΔT: 100ms

Control output cycle: CTDUTY = 2.0s

Input: MV0 = 25%, MV1 = 35%, MV2 = 40%, MV3 = 35%

Output ON delay time: DELAY = 0.1s

Start FB specification: PRIMARY = TRUE





- Changing nine or more bit ON/OFF duty output phases

If multiple P\_DUTY\_8PT\_ function blocks are connected, nine or more bit ON/OFF duty output phases can be changed.

When connecting multiple P\_DUTY\_8PT\_ function blocks, connect LINKOUT of the preceding FB with LINKIN of the following FB, set PRIMARY of the preceding FB to TRUE, and PRIMARY of the following FB to FALSE. The value of CTDUTY and DELAY of the first FB is applied to the value of CTDUTY and DELAY of the following FB.

**Ex.**

Execution cycle:  $\Delta T$ : 100ms

- First FB

Control output cycle: CTDUTY = 2.0s

Input: MV0 = 40%, MV1 = 20%, MV2 = 40%, MV3 = 15%, MV4 = 30%, MV5 = 100%, MV6 = 0%, MV7 = 35%

Output ON delay time: DELAY = 0.1s

Lead FB specified: PRIMARY = TRUE

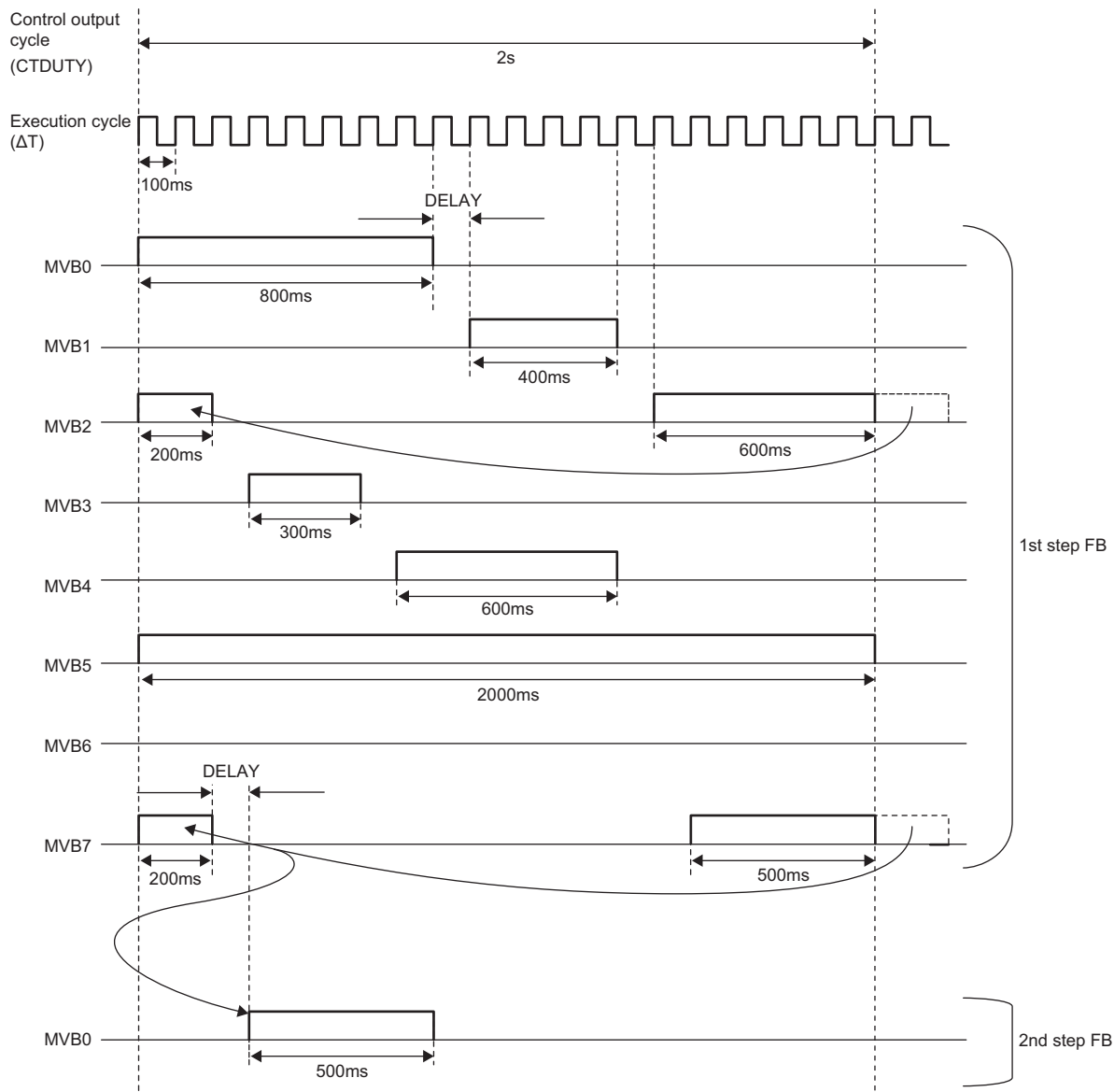
- Second FB

Control output cycle: CTDUTY = Same as CTDUTY of the preceding FB (2.0s in this example)

Input: MV0 = 25%

Output ON delay time: DELAY = Same as DELAY of the preceding FB (0.1s in this example)

Start FB specification: PRIMARY = FALSE



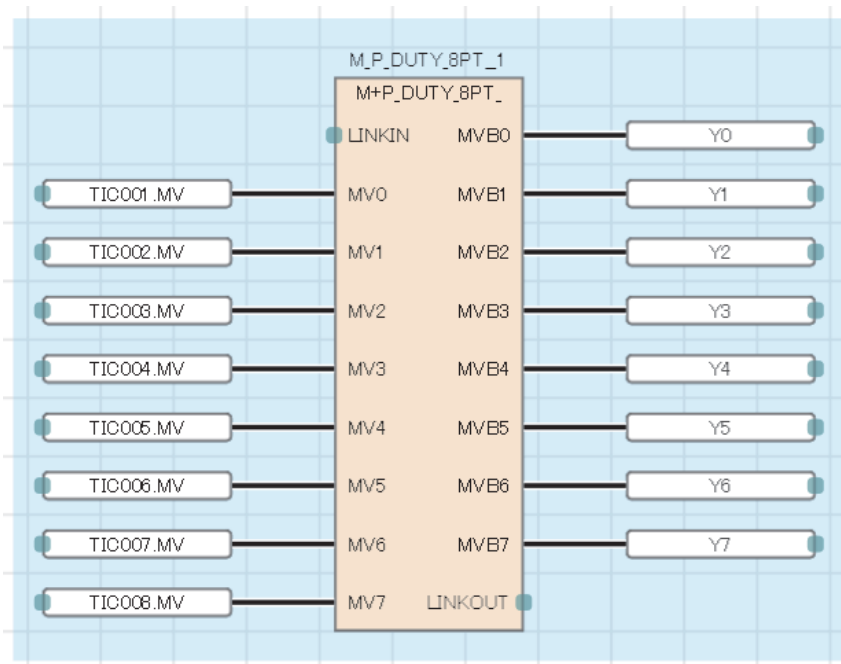
## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (MV0 to MV7) or operation constant is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

## Program example

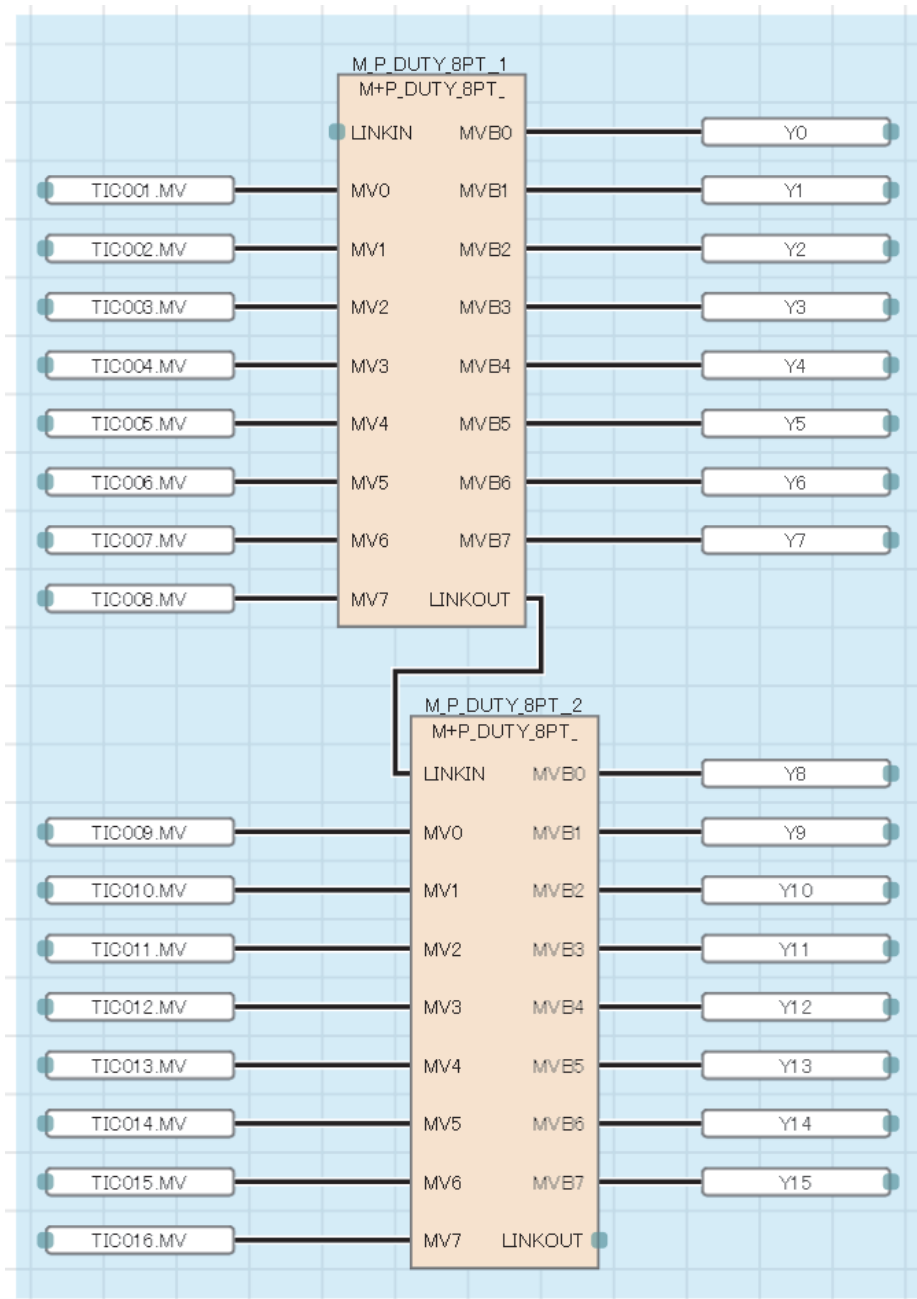
- When one M+P\_DUTY\_8PT\_ is used

Variable type/pin	Variable name	Description	Setting/connection method
Public variable	PRIMARY	Lead FB specified	TRUE
Input pin	LINKIN	Link input	Not connected
Output pin	LINKOUT	Link output	Not connected



- When multiple M+P\_DUTY\_8PT\_ function blocks are connected (when nine or more bit ON/OFF duty output phases is changed)

Target FB	Variable type/pin	Variable name	Description	Setting/connection method
Start FB	Public variable	PRIMARY	Lead FB specified	TRUE
	Input pin	LINKIN	Link input	Not connected
	Output pin	LINKOUT	Link output	Connected with LINKIN of the following FB
Following FB	Public variable	PRIMARY	Lead FB specified	FALSE
	Input pin	LINKIN	Link input	Connected with LINKOUT of the preceding FB
	Output pin	LINKOUT	Link output	Connected with LINKIN of the following FB
Last FB	Public variable	PRIMARY	Lead FB specified	FALSE
	Input pin	LINKIN	Link input	Connected with LINKOUT of the preceding FB
	Output pin	LINKOUT	Link output	Not connected



**Point** 

---

When multiple P\_DUTY\_8PT\_ function blocks are connected, the control output cycle and output ON delay time of the following FB used in operations can be checked with the value of the public variables CTDUTY and DELAY of the start FB.

---

# 10 BISTABLE SWITCH

The following function blocks set/reset priorities and output the result of latch processing.

**Point** The output variables and public variables of general process FBs are latched.

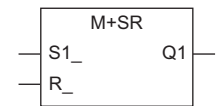
**Point** To set initial values of public variables of the general process FB arranged on a user-defined tag FB in "FB Property" of the engineering tool, refer to the following.  
 📖 GX Works3 Operating Manual

## 10.1 Bistable Switch (Set Dominant) (M+SR)

### M+SR

This function block discriminates between two input values, and outputs 1 (TRUE) or 0 (FALSE).

#### FBD/LD



#### Setting data

##### Input/output variable

Variable name	Description	Recommended range	Type	Data type
S1_	Set Command	TRUE, FALSE	Input variable	BOOL
R_	Reset Command	TRUE, FALSE	Input variable	BOOL
Q1	Output	TRUE, FALSE	Output variable	BOOL

##### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IR	Reset Request at CPU Initial Execution	TRUE: After the start-up of the CPU module (powered off and on, reset), when an FB instance is called for the first time, this function block resets the output value in the output variable Q1. FALSE: After the start-up of the CPU module (powered off and on, reset), when an FB instance is called for the first time, this function block holds the output value in the output variable Q1.	FALSE	User	BOOL

## Processing details

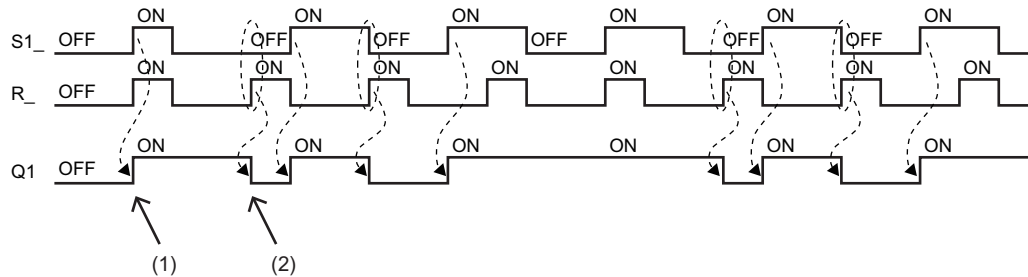
### ■ Operation processing

When the input variable S1\_ turns on, this function block sets the output variable Q1. While the input variable S1\_ is off, when the input variable R\_ is turned on, the output variable Q1 can be reset.

While the input variable S1\_ is on, even when the input variable R\_ is turned on, the output variable will not be reset.

When the public variable IR is TRUE, at the start-up of the CPU module (powered off and on, reset), this function block resets the output value in the output variable Q1. When the public variable IR is FALSE, this function block holds the output value in the output variable Q1.

### ■ Operation result



(1) When S1\_ = on, Q1 is set.

(2) When S1\_ = off and R\_ = on, Q1 is reset.

## Operation error

There is no operation error in M+SR.

### Point

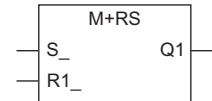
- To set initial values of public variables of the general process FB arranged on a user-defined FB/tag FB in the FB Property window or read/write them in a program, the public variables must be referred to on the user-defined FB/tag FB.
- For use with the public variable IR (Reset Request at CPU Initial Execution) being TRUE, do not refer to the output variable Q1 of an FB instance before FB execution. Otherwise, the value before reset may be obtained. To refer to the output variable Q1 before FB execution, refer to a program example of matching operations with the operations of PX Developer. (Page 947 Program Example of Matching Operations with the Operations of PX Developer)

## 10.2 Bistable Switch (Reset Dominant) (M+RS)

### M+RS

This function block discriminates between two input values, and outputs 1 (TRUE) or 0 (FALSE).

#### FBD/LD



10

### Setting data

#### ■ Input/output variable

Variable name	Description	Recommended range	Type	Data type
S_	Set Command	TRUE, FALSE	Input variable	BOOL
R1_	Reset Command	TRUE, FALSE	Input variable	BOOL
Q1	Output	TRUE, FALSE	Output variable	BOOL

#### ■ Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IR	Reset Request at CPU Initial Execution	TRUE: After the start-up of the CPU module (powered off and on, reset), when an FB instance is called for the first time, this function block resets the output value in the output variable Q1. FALSE: After the start-up of the CPU module (powered off and on, reset), when an FB instance is called for the first time, this function block holds the output value in the output variable Q1.	FALSE	User	BOOL

## Processing details

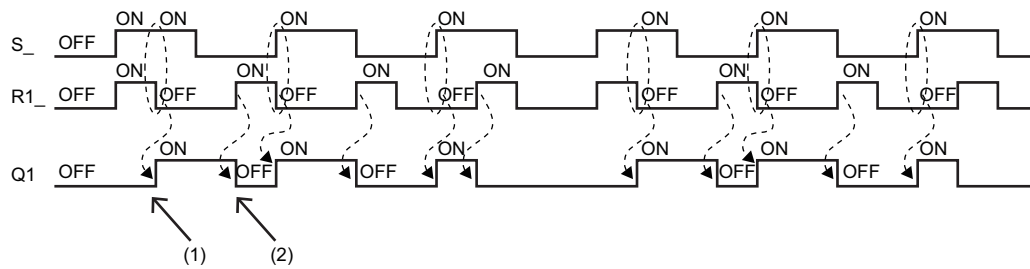
### ■ Operation processing

When the input variable S\_ turns on, this function block sets the output variable Q1. When the input variable R1\_ is turned on, this function block resets the output variable Q1.

While the input variable R1\_ is on, even when the input variable S\_ is turned on, the output variable Q1 cannot be set.

When the public variable IR is TRUE, at the start-up of the CPU module (powered off and on, reset), this function block resets the output value in the output variable Q1. When the public variable IR is FALSE, this function block holds the output value in the output variable Q1.

### ■ Operation result



(1) When S\_ = on and R1\_ = off, Q1 is set.

(2) When R1\_ = on, Q1 is reset.

## Operation error

There is no operation error in M+RS.

### Point

- To set initial values of public variables of the general process FB arranged on a user-defined FB/tag FB in the FB Property window or read/write them in a program, the public variables must be referred to on the user-defined FB/tag FB.
- For use with the public variable IR (Reset Request at CPU Initial Execution) being TRUE, do not refer to the output variable Q1 of an FB instance before FB execution. Otherwise, the value before reset may be obtained. To refer to the output variable Q1 before FB execution, refer to a program example of matching operations with the operations of PX Developer. (☞ Page 947 Program Example of Matching Operations with the Operations of PX Developer)



# 10.3 Latch FB (BOOL Type) (M+LATCH\_BOOL)

## M+LATCH\_BOOL

This function block outputs the result of latch processing. (BOOL type)

### FBD/LD



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
E_	Latch Processing	TRUE: Not latch FALSE: Latch	Input variable	BOOL
IN	Input	TRUE, FALSE	Input variable	BOOL
OUT_	Output	TRUE, FALSE	Output variable	BOOL

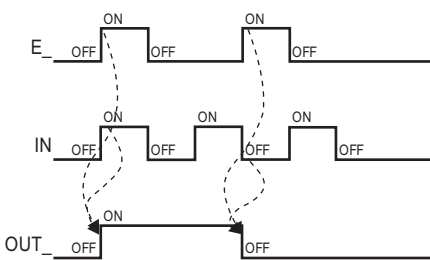
### Processing details

#### Operation processing

When the input variable E\_ is on, this function block outputs the input variable IN to the output variable OUT\_. When the input variable E\_ is off, the output variable OUT\_ holds the last value regardless of the input of the input variable IN.

#### Operation result

Input variable		Output variable
E_ (BOOL type)	IN (BOOL type)	OUT_ (BOOL type)
TRUE	TRUE	TRUE
	FALSE	FALSE
FALSE	TRUE	Last output value
	FALSE	



### Operation error

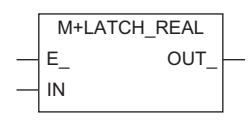
There is no operation error in M+LATCH\_BOOL.

# 10.4 Latch FB (REAL Type) (M+LATCH\_REAL)

## M+LATCH\_REAL

This function block outputs the result of latch processing. (REAL type)

### FBD/LD



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
E_	Latch Processing	TRUE: Not latch FALSE: Latch	Input variable	BOOL
IN	Input	$-2^{128}$ to $-2^{-126}$ , 0, $2^{-126}$ to $2^{128}$	Input variable	REAL
OUT_	Output	$-2^{128}$ to $-2^{-126}$ , 0, $2^{-126}$ to $2^{128}$	Output variable	REAL

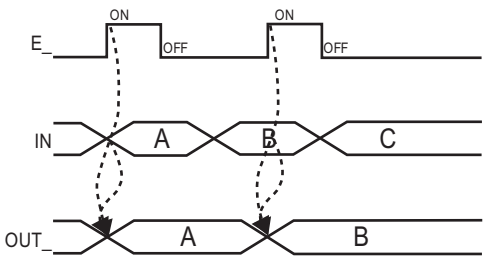
### Processing details

#### Operation processing

When the input variable E\_ is on, this function block outputs the input variable IN to the output variable OUT\_. When the input variable E\_ is off, the output variable OUT\_ holds the last value regardless of the input of the input variable IN.

#### Operation result

Input variable		Output variable
E_ (BOOL type)	IN (REAL type)	OUT_ (REAL type)
TRUE	Input value	Input value
FALSE	Input value	Last value



### Operation error

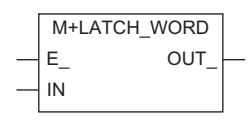
There is no operation error in M+LATCH\_REAL.

# 10.5 Latch FB (WORD Type) (M+LATCH\_REAL)

## M+LATCH\_WORD

This function block outputs the result of latch processing. (WORD type)

### FBD/LD



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
E_	Latch Processing	TRUE: Not latch FALSE: Latch	Input variable	BOOL
IN	Input	0 to 65535	Input variable	WORD
OUT_	Output	0 to 65535	Output variable	WORD

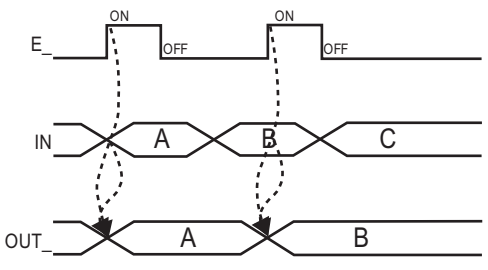
### Processing details

#### Operation processing

When the input variable E\_ is on, this function block outputs the input variable IN to the output variable OUT\_. When the input variable E\_ is off, the output variable OUT\_ holds the last value regardless of the input of the input variable IN.

#### Operation result

Input variable		Output variable
E_ (BOOL type)	IN (WORD type)	OUT_ (WORD type)
TRUE	Input value	Input value
FALSE	Input value	Last value



### Operation error

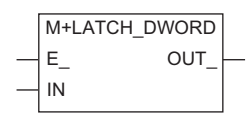
There is no operation error in M+LATCH\_WORD.

# 10.6 Latch FB (DWORD Type) (M+LATCH\_DWORD)

## M+LATCH\_DWORD

This function block outputs the result of latch processing. (DWORD type)

### FBD/LD



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
E_	Latch Processing	TRUE: Not latch FALSE: Latch	Input variable	BOOL
IN	Input	0 to 4294967295	Input variable	DWORD
OUT_	Output	0 to 4294967295	Output variable	DWORD

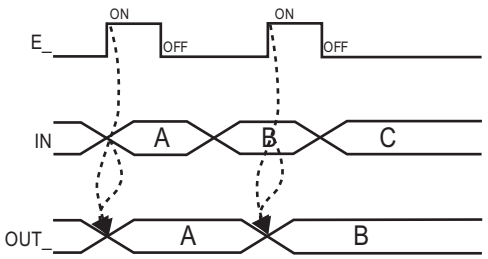
### Processing details

#### Operation processing

When the input variable E\_ is on, this function block outputs the input variable IN to the output variable OUT\_. When the input variable E\_ is off, the output variable OUT\_ holds the last value regardless of the input of the input variable IN.

#### Operation result

Input variable		Output variable
E_ (BOOL type)	IN (DWORD type)	OUT_ (DWORD type)
TRUE	Input value	Input value
FALSE	Input value	Last value



### Operation error

There is no operation error in M+LATCH\_DWORD.

# 11 EDGE DETECTOR

The following function blocks produce rising/falling edge detection outputs.



The output variables and public variables of general process FBs are latched.



To set initial values of public variables of the general process FB arranged on a user-defined tag FB in "FB Property" of the engineering tool, refer to the following.

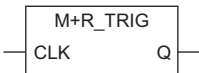
GX Works3 Operating Manual

## 11.1 Rising Edge Detection (M+R\_TRIG)

### M+R\_TRIG

This function block detects a signal rising edge and outputs a pulse signal.

#### FBD/LD



#### Setting data

##### Input/output variable

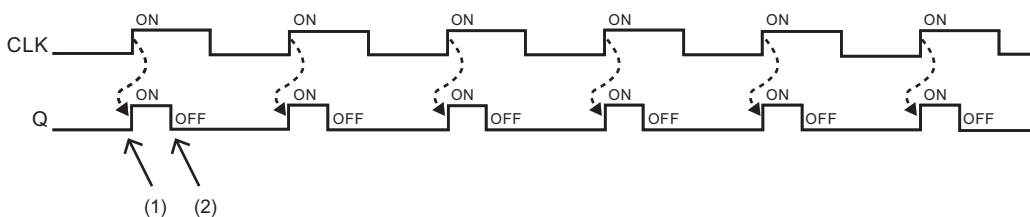
Variable name	Description	Recommended range	Type	Data type
CLK	Rising Edge Detection	TRUE, FALSE	Input variable	BOOL
Q	Output	TRUE, FALSE	Output variable	BOOL

#### Processing details

##### Operation processing

When the input variable CLK turns on, this function block turns on the output variable Q only for one scan.

##### Operation result



- (1) On the rising edge of CLK, Q is turned on.
- (2) Q is turned off at the next scan.

#### Operation error

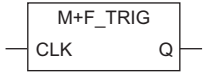
There is no operation error in M+R\_TRIG.

# 11.2 Falling Edge Detection (M+F\_TRIG)

## M+F\_TRIG

This function block detects a signal falling edge and outputs a pulse signal.

### FBD/LD



### Setting data

#### Input/output variable

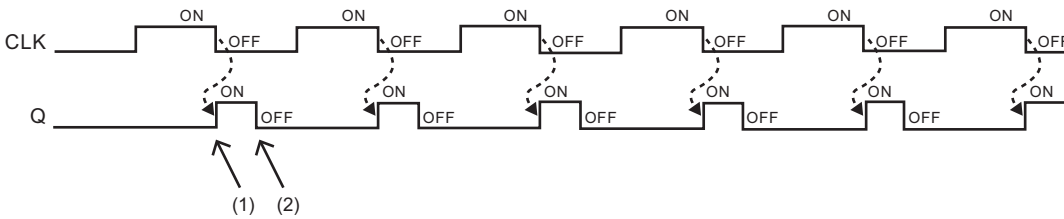
Variable name	Description	Recommended range	Type	Data type
CLK	Falling Edge Detection	TRUE, FALSE	Input variable	BOOL
Q	Output	TRUE, FALSE	Output variable	BOOL

### Processing details

#### Operation processing

When the input variable CLK turns off, this function block turns on the output variable Q only for one scan.

#### Operation result



- (1) On the falling edge of CLK, Q is turned on.
- (2) Q is turned off at the next scan.

### Operation error

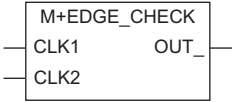
There is no operation error in M+F\_TRIG.

# 11.3 Edge Detector (M+EDGE\_CHECK)

## M+EDGE\_CHECK

This function block outputs a rising edge detection or a falling edge detection.  
 (When the input variable CLK1 turns off and on, and the input variable CLK2 turns on and off)

### FBD/LD



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
CLK1	Rising Edge Detection	TRUE, FALSE	Input variable	BOOL
CLK2	Falling Edge Detection	TRUE, FALSE	Input variable	BOOL
OUT_	Output	TRUE, FALSE	Output variable	BOOL

### Processing details

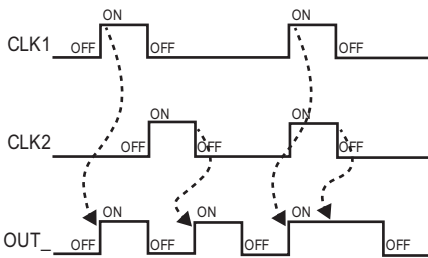
#### Operation processing

When a rising edge of the input variable CLK1 or a falling edge of the input variable CLK2 is detected, the output variable OUT\_ turns on only for one scan.

#### Operation result

When the input value of the input variable CLK1 or CLK2 is as follows, the output variable OUT\_ turns off.

- Input value of the input variable CLK1: Not off to on (i.e. on to on, on to off, off to off)
- Input value of the input variable CLK2: Not on to off (i.e. off to off, off to on, on to on).



### Operation error

There is no operation error in M+EDGE\_CHECK.

# 12 COUNTER


The following function blocks produce outputs of current value + 1 (addition) and current value - 1 (subtraction).

## Point

The output variables and public variables of general process FBs are latched.

## Point

To set initial values of public variables of the general process FB arranged on a user-defined tag FB in "FB Property" of the engineering tool, refer to the following.

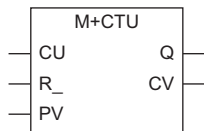
 GX Works3 Operating Manual

## 12.1 Up Counter (M+CTU)

### M+CTU

This function block counts up the number of rising edges of a signal.

#### FBD/LD



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
CU	Rising Edge Detection	TRUE, FALSE	Input variable	BOOL
R_	Reset Command	TRUE, FALSE	Input variable	BOOL
PV	Counter Maximum Value	0 to 32767	Input variable	INT
Q	Count Completed	TRUE, FALSE	Output variable	BOOL
CV	Count Value	0 to 32767	Output variable	INT

### Processing details

#### Operation processing

[Up count]

When the input variable CU turns off and on, this function block counts up the output variable CV by one.

When the output variable CV reaches PV (Counter Maximum Value), the output variable Q turns on and up count stops.

Set the maximum value of the counter to the input variable PV. When the input variable R\_ is turned on, the output variable Q turns off and the output variable CV is set to 0.

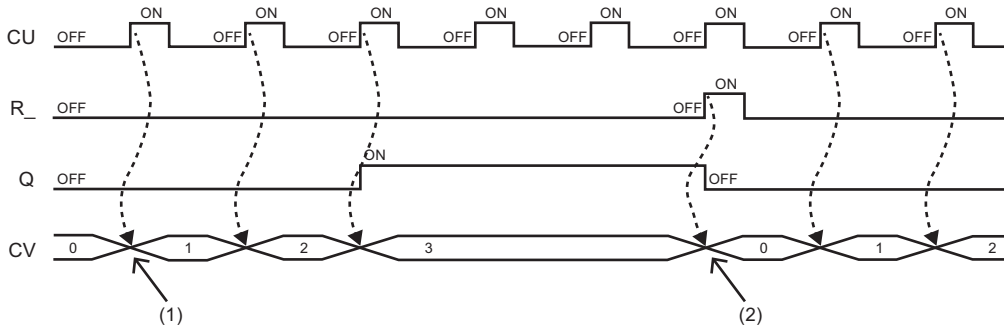
[Counter Maximum Value]

The valid setting range for the input variable PV is 0 to 32767.



## ■ Operation result

When PV = 3



- (1) When CU = on, CV is counted up.
- (2) When R\_ = on, CV is cleared to 0.

## Operation error

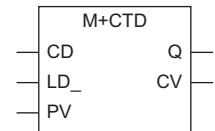
There is no operation error in M+CTU.

# 12.2 Down Counter (M+CTD)

## M+CTD

This function block counts down the number of rising edges of a signal.

### FBD/LD



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
CD	Rising Edge Detection	TRUE, FALSE	Input variable	BOOL
LD_	Load Command	TRUE, FALSE	Input variable	BOOL
PV	Counter Start Value	0 to 32767	Input variable	INT
Q	Count Completed	TRUE, FALSE	Output variable	BOOL
CV	Count Value	0 to 32767	Output variable	INT

### Processing details

#### Operation processing

[Down count]

When the input variable CD turns off and on, this function block counts down the output variable CV by one.

When the value of the output variable CV is 0, the output variable Q turns on and down count stops.

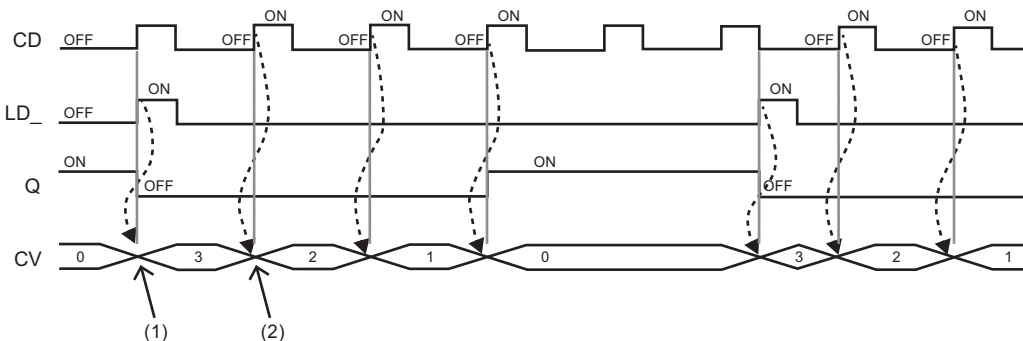
Set the start value of the counter to the input variable PV. When the input variable LD\_ is turned on, the output variable Q turns off and PV (Counter Start Value) is set to the output variable CV.

[Counter Start Value]

The valid setting range for the input variable PV is 0 to 32767.

#### Operation result

When PV = 3



- (1) When LD\_ = on, CV is initialized with the value in PV.
- (2) On the rising edge of CD, CV is counted down.

### Operation error

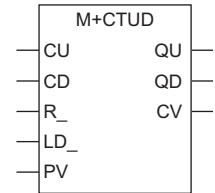
There is no operation error in M+CTD.

# 12.3 Up/Down Counter (M+CTUD)

## M+CTUD

This function block counts up or down the number of rising edges of a signal.

### FBD/LD



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
CU	Rising Edge Detection (Up Counter)	TRUE, FALSE	Input variable	BOOL
CD	Rising Edge Detection (Down Counter)	TRUE, FALSE	Input variable	BOOL
R_	Reset Command	TRUE, FALSE	Input variable	BOOL
LD_	Load Command	TRUE, FALSE	Input variable	BOOL
PV	Counter Maximum Value/Start Value	0 to 32767	Input variable	INT
QU	Count Completed (Up Counter)	TRUE, FALSE	Output variable	BOOL
QD	Count Completed (Down Counter)	TRUE, FALSE	Output variable	BOOL
CV	Count Value	0 to 32767	Output variable	INT

### Processing details

#### Operation processing

[Up count]

When the input variable CU turns off and on, this function block counts up the output variable CV by one.

When the output variable CV reaches the maximum value of the counter set in PV, the output variable QU turns on and up count stops.

Set the maximum value of the counter to the input variable PV. When the input variable R\_ is turned on, the output variable QU turns off and the output variable CV is set to 0.

[Down count]

When the input variable CD turns off and on, this function block counts down the output variable CV by one.

When the value of the output variable CV is 0, the output variable QD turns on and down count stops.

Set the start value of the counter to the input variable PV. When the input variable LD\_ is turned on, the output variable QD turns off and the start value of the counter set in PV is set to the output variable CV.

[Counter Maximum Value/Start Value]

The valid setting range for the input variable PV is 0 to 32767.

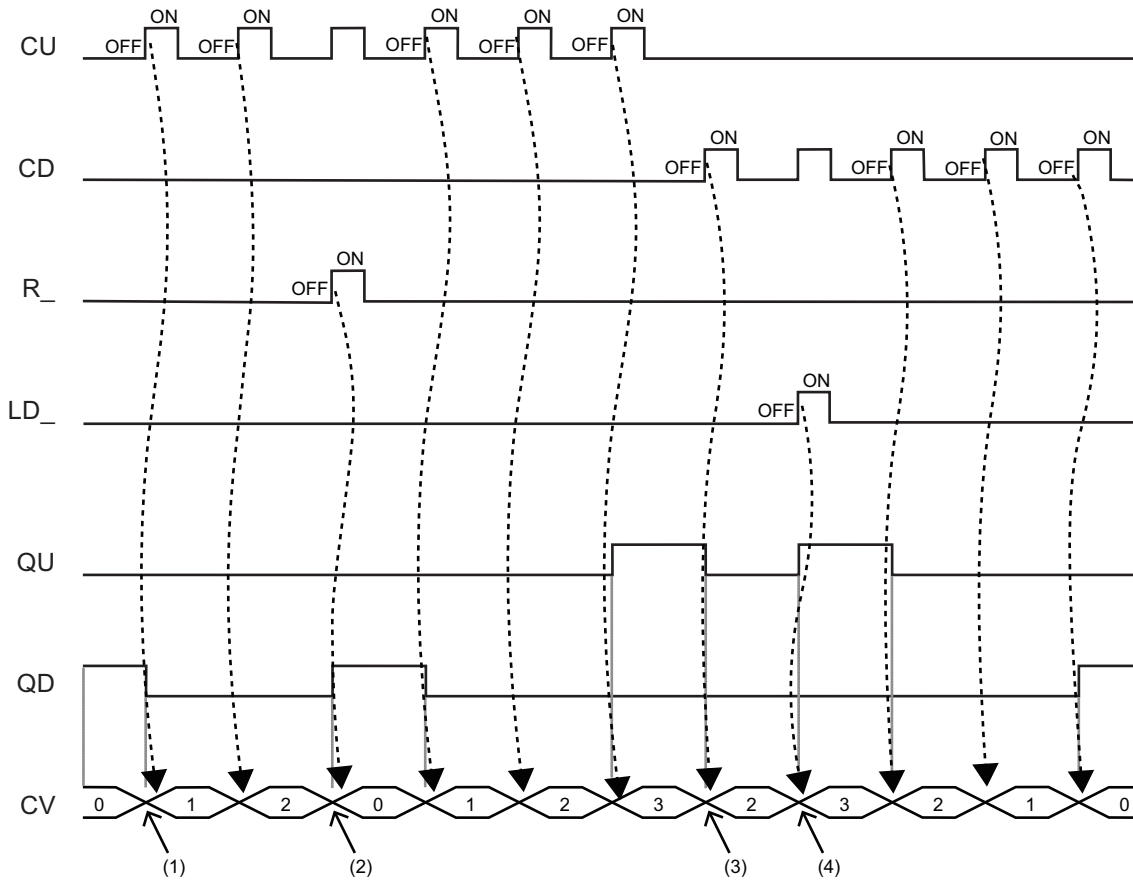
[Other processing]

When the input variables CU and CD turn off and on simultaneously, priority is given to the input variable CU and this function block counts up the output variable CV by one.

When the input variables R\_ and LD\_ are turned on simultaneously, priority is given to the input variable R\_ and this function block sets the output variable CV to 0.

## ■ Operation result

When PV = 3



- (1) When CU turns off and on, CV is counted up.
- (2) When R\_ turns off and on, CV is initialized.
- (3) When CD turns off and on, CV is counted down.
- (4) When LD\_ turns off and on, CV is initialized.

## Operation error

There is no operation error in M+CTUD.

# 13 TIMER


The following function blocks process the pulse timer, ON delay timer, and OFF delay timer.

## Point

The output variables and public variables of general process FBs are latched.

## Point

To set initial values of public variables of the general process FB arranged on a user-defined tag FB in "FB Property" of the engineering tool, refer to the following.

 GX Works3 Operating Manual

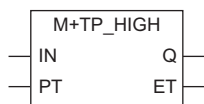
## 13.1 Pulse Timer (High-speed Timer Type) (M+TP\_HIGH)

13

### M+TP\_HIGH

This function block keeps the signal on for the specified period of time.

#### FBD/LD



## Point

A project using this FB cannot be opened with a version earlier than 1.080J of the engineering tool.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN	Rising Edge Detection	TRUE, FALSE	Input variable	BOOL
PT	Pulse Width Setting	1 to 32767	Input variable	INT
Q	Output	TRUE, FALSE	Output variable	BOOL
ET	Total Measuring Time	0 to 32767	Output variable	INT

## Processing details

### ■ Operation processing

[Output]

When the input variable IN turns on, this function block keeps the output variable Q on for the period of time set in the input variable PT.

The time elapsed after the output variable Q turns on is set to the output variable ET.

Use the long timer to count the elapsed time.

[End of output]

Once the elapsed time reaches the set time, this function block turns off the output variable Q.

After Q is turned off, if the input variable IN is turned off, this function block resets the elapsed time.

Even when the input variable IN turns off while the output variable Q is on, Q is not turned off.

[Pulse width setting]

The pulse width time [ms] is input variable PT × high-speed timer setting value [ms] in the timer limit setting<sup>\*1</sup>.

\*1 The value set in [Navigation window] ⇒ [Parameter] ⇒ CPU module ⇒ [CPU Parameter] ⇒ [Operation Related Setting] ⇒ [Timer Limit Setting] ⇒ [High Speed Timer/High Speed Retentive Timer]

The valid setting range for the pulse width time [ms] is determined by the long timer setting value in the timer limit setting<sup>\*2</sup>.

However, since the data type of the pulse width time is TIME data type (32-bit value), the maximum value will fall within the range of TIME data type.

\*2 The value set in [Navigation window] ⇒ [Parameter] ⇒ CPU module ⇒ [CPU Parameter] ⇒ [Operation Related Setting] ⇒ [Timer Limit Setting] ⇒ [Long Timer/Long Retentive Timer]

- Minimum value

Identical to the long timer setting value [ms] in the timer limit setting.

Note that if the long timer setting value is smaller than 1ms, the minimum value will be 1ms.

- Maximum value

Pulse width value [ms] ≤ 2147483647 [ms] × long timer setting value [ms] in the timer limit setting

**Ex.**

If the long timer setting value is 0.001ms: T#1ms to T#2147400ms

If the long timer setting value is 1000ms: T#1000ms to T#32767000ms

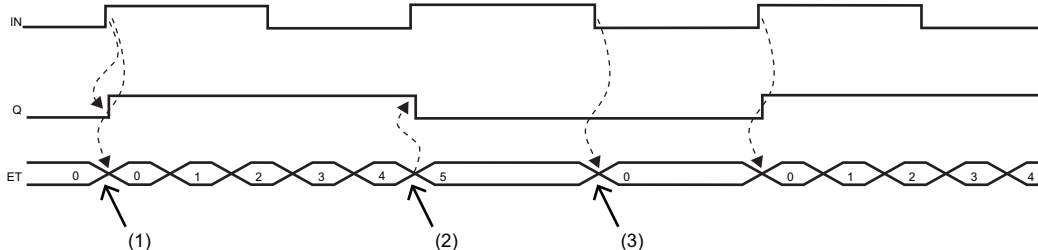
The value when the output variable Q turns off and on (rising edge) is used for the setting value of the input variable PT.

When the value of the input variable PT is changed while the output variable Q is on, the new value will be enabled at the next output start timing.

### ■ Operation result

Operation result	Q, ET
No operation error	Operation result output value
Operation error	Undefined value

When PT = 500 (when the high-speed timer setting value in the timer limit setting is 10ms) (5 seconds)



(1) When IN = on, Q is turned on. When IN = on, ET starts measuring time.

(2) When the time measured in ET reaches the time set in PT, Q is turned off.

(3) When Q1 = off and IN = off, ET is initialized.

## Operation error

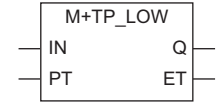
Error code (SD0)	Description
3401H	The pulse width time [ms] exceeds the valid range.

# 13.2 Pulse Timer (Low-speed Timer Type) (M+TP\_LOW)

## M+TP\_LOW

This function block keeps the signal on for the specified period of time.

### FBD/LD



A project using this FB cannot be opened with a version earlier than 1.080J of the engineering tool.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN	Rising Edge Detection	TRUE, FALSE	Input variable	BOOL
PT	Pulse Width Setting	1 to 32767	Input variable	INT
Q	Output	TRUE, FALSE	Output variable	BOOL
ET	Total Measuring Time	0 to 32767	Output variable	INT

## Processing details

### ■ Operation processing

[Output]

When the input variable IN turns on, this function block keeps the output variable Q on for the period of time set in the input variable PT.

The time elapsed after the output variable Q turns on is set to the output variable ET.

Use the long timer to count the elapsed time.

[End of output]

Once the elapsed time reaches the set time, this function block turns off the output variable Q.

After Q is turned off, if the input variable IN is turned off, this function block resets the elapsed time.

Even when the input variable IN turns off while the output variable Q is on, Q is not turned off.

[Pulse width setting]

The pulse width time [ms] is input variable PT × low-speed timer setting value [ms] in the timer limit setting<sup>\*1</sup>.

\*1 The value set in [Navigation window] ⇒ [Parameter] ⇒ CPU module ⇒ [CPU Parameter] ⇒ [Operation Related Setting] ⇒ [Timer Limit Setting] ⇒ [Low Speed Timer/Low Speed Retentive Timer]

The valid setting range for the pulse width time [ms] is determined by the long timer setting value in the timer limit setting<sup>\*2</sup>.

However, since the data type of the pulse width time is TIME data type (32-bit value), the maximum value will fall within the range of TIME data type.

\*2 The value set in [Navigation window] ⇒ [Parameter] ⇒ CPU module ⇒ [CPU Parameter] ⇒ [Operation Related Setting] ⇒ [Timer Limit Setting] ⇒ [Long Timer/Long Retentive Timer]

- Minimum value

Identical to the long timer setting value [ms] in the timer limit setting.

Note that if the long timer setting value is smaller than 1ms, the minimum value will be 1ms.

- Maximum value

Pulse width value [ms] ≤ 2147483647 [ms] × long timer setting value [ms] in the timer limit setting

**Ex.**

If the long timer setting value is 0.001ms: T#1ms to T#2147400ms

If the long timer setting value is 1000ms: T#1000ms to T#32767000ms

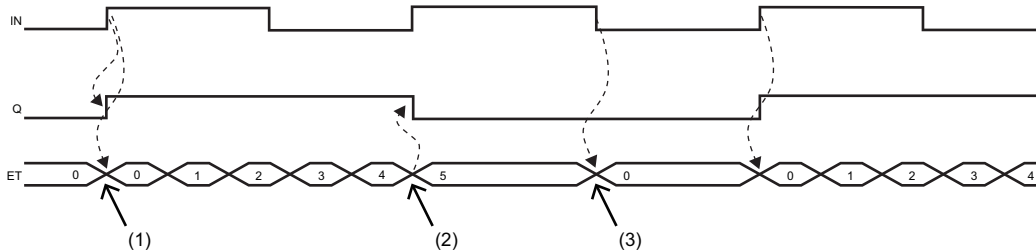
The value when the output variable Q turns off and on (rising edge) is used for the setting value of the input variable PT.

When the value of the input variable PT is changed while the output variable Q is on, the new value will be enabled at the next output start timing.

### ■ Operation result

Operation result	Q, ET
No operation error	Operation result output value
Operation error	Undefined value

When PT = 500 (when the low-speed timer setting value in the timer limit setting is 10ms) (5 seconds)



(1) When IN = on, Q is turned on. When IN = on, ET starts measuring time.

(2) When the time measured in ET reaches the time set in PT, Q is turned off.

(3) When Q1 = off and IN = off, ET is initialized.

## Operation error

Error code (SD0)	Description
3401H	The pulse width time [ms] exceeds the valid range.

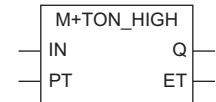


# 13.3 ON Delay Timer (High-speed Timer Type) (M+TON\_HIGH)

## M+TON\_HIGH

This function blocks turns on a signal after the specified period of time.

### FBD/LD



### Point

A project using this FB cannot be opened with a version earlier than 1.080J of the engineering tool.

13

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN	Rising Edge Detection	TRUE, FALSE	Input variable	BOOL
PT	ON Delay Timer Setting	1 to 32767	Input variable	INT
Q	Output	TRUE, FALSE	Output variable	BOOL
ET	Measuring Current Value	0 to 32767	Output variable	INT

## Processing details

### ■ Operation processing

[Output]

When the input variable IN turns on, this function block turns on the output variable Q after the period of time set in the input variable PT.

The delay time elapsed until the output variable Q turns on is set to the output variable ET.

When the input variable IN turns off, this function block turns off the output variable Q and resets the elapsed delay time.

Use the long timer to count the elapsed time.

[ON delay timer setting]

The ON delay timer time [ms] is input variable PT × high-speed timer setting value [ms] in the timer limit setting<sup>\*1</sup>.

\*1 The value set in [Navigation window] ⇒ [Parameter] ⇒ CPU module ⇒ [CPU Parameter] ⇒ [Operation Related Setting] ⇒ [Timer Limit Setting] ⇒ [High Speed Timer/High Speed Retentive Timer]

The valid setting range for the ON delay timer time [ms] is determined by the long timer setting value in the timer limit setting<sup>\*2</sup>.

However, since the data type of the ON delay timer time is TIME data type (32-bit value), the maximum value will fall within the range of TIME data type.

\*2 The value set in [Navigation window] ⇒ [Parameter] ⇒ CPU module ⇒ [CPU Parameter] ⇒ [Operation Related Setting] ⇒ [Timer Limit Setting] ⇒ [Long Timer/Long Retentive Timer]

- Minimum value

Identical to the long timer setting value [ms] in the timer limit setting.

Note that if the long timer setting value is smaller than 1ms, the minimum value will be 1ms.

- Maximum value

ON delay timer time [ms] ≤ 2147483647 [ms] × long timer setting value [ms] in the timer limit setting

**Ex.**

If the long timer setting value is 0.001ms: T#1ms to T#2147400ms

If the long timer setting value is 1000ms: T#1000ms to T#32767000ms

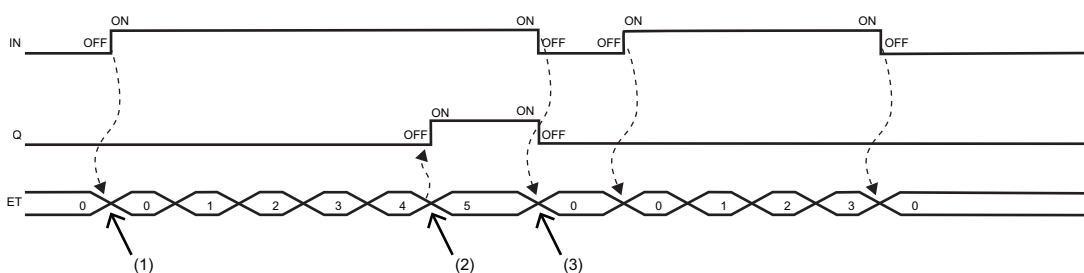
The value when the output variable Q turns off and on (rising edge) is used for the setting value of the input variable PT.

When the value of the input variable PT is changed while the input variable IN is on, the new value will be enabled at the next rising edge of IN.

### ■ Operation result

Operation result	Q, ET
No operation error	Operation result output value
Operation error	Undefined value

When PT = 500 (when the high-speed timer setting value in the timer limit setting is 10ms) (5 seconds)



- (1) When IN = on, ET starts measuring time.
- (2) When the time measured in ET reaches the time set in PT, Q is turned on.
- (3) On the rising edge of IN, Q is turned off and ET is reset.

## Operation error

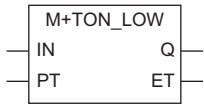
Error code (SD0)	Description
3401H	The ON delay timer time [ms] exceeds the valid range.

# 13.4 ON Delay Timer (Low-speed Timer Type) (M+TON\_LOW)

## M+TON\_LOW

This function block turns on a signal after the specified period of time.

FBD/LD



### Point

A project using this FB cannot be opened with a version earlier than 1.080J of the engineering tool.

13

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN	Rising Edge Detection	TRUE, FALSE	Input variable	BOOL
PT	ON Delay Timer Setting	1 to 32767	Input variable	INT
Q	Output	TRUE, FALSE	Output variable	BOOL
ET	Measuring Current Value	0 to 32767	Output variable	INT

## Processing details

### ■ Operation processing

[Output]

When the input variable IN turns on, this function block turns on the output variable Q after the period of time set in the input variable PT.

The delay time elapsed until the output variable Q turns on is set to the output variable ET.

When the input variable IN turns off, this function block turns off the output variable Q and resets the elapsed delay time.

Use the long timer to count the elapsed time.

[ON delay timer setting]

The ON delay timer time [ms] is input variable PT × low-speed timer setting value [ms] in the timer limit setting<sup>\*1</sup>.

\*1 The value set in [Navigation window] ⇒ [Parameter] ⇒ CPU module ⇒ [CPU Parameter] ⇒ [Operation Related Setting] ⇒ [Timer Limit Setting] ⇒ [Low Speed Timer/Low Speed Retentive Timer]

The valid setting range for the ON delay timer time [ms] is determined by the long timer setting value in the timer limit setting<sup>\*2</sup>.

However, since the data type of the ON delay timer time is TIME data type (32-bit value), the maximum value will fall within the range of TIME data type.

\*2 The value set in [Navigation window] ⇒ [Parameter] ⇒ CPU module ⇒ [CPU Parameter] ⇒ [Operation Related Setting] ⇒ [Timer Limit Setting] ⇒ [Long Timer/Long Retentive Timer]

- Minimum value

Identical to the long timer setting value [ms] in the timer limit setting.

Note that if the long timer setting value is smaller than 1ms, the minimum value will be 1ms.

- Maximum value

ON delay timer time [ms] ≤ 2147483647 [ms] × long timer setting value [ms] in the timer limit setting

**Ex.**

If the long timer setting value is 0.001ms: T#1ms to T#2147400ms

If the long timer setting value is 1000ms: T#1000ms to T#32767000ms

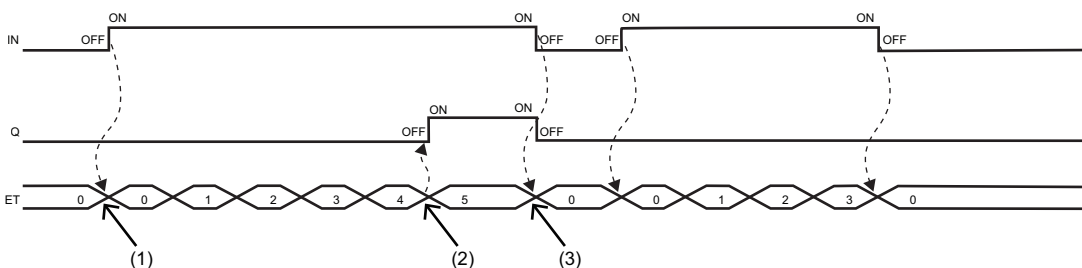
The value when the output variable Q turns off and on (rising edge) is used for the setting value of the input variable PT.

When the value of the input variable PT is changed while the input variable IN is on, the new value will be enabled at the next rising edge of IN.

### ■ Operation result

Operation result	Q, ET
No operation error	Operation result output value
Operation error	Undefined value

When PT = 500 (when the low-speed timer setting value in the timer limit setting is 10ms) (5 seconds)



- (1) When IN = on, ET starts measuring time.
- (2) When the time measured in ET reaches the time set in PT, Q is turned on.
- (3) On the rising edge of IN, Q is turned off and ET is reset.

## Operation error

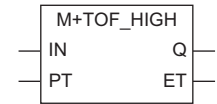
Error code (SD0)	Description
3401H	The ON delay timer time [ms] exceeds the valid range.

# 13.5 OFF Delay Timer (High-speed Timer Type) (M+TOF\_HIGH)

## M+TOF\_HIGH

This function block turns off a signal after the specified period of time.

### FBD/LD



### Point

A project using this FB cannot be opened with a version earlier than 1.080J of the engineering tool.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN	Rising Edge Detection	TRUE, FALSE	Input variable	BOOL
PT	OFF Delay Timer Setting	1 to 32767	Input variable	INT
Q	Output	TRUE, FALSE	Output variable	BOOL
ET	Measuring Current Value	0 to 32767	Output variable	INT

## Processing details

### ■ Operation processing

[Output]

When the input variable IN turns on, this function turns on the output variable Q.

When the input variable IN turns on and off, this function block turns off the output variable Q after the period of time set in the input variable PT.

The time elapsed until the output variable Q turns off is set to the output variable ET.

Use the long timer to count the elapsed time.

[OFF delay timer setting]

The OFF delay timer time [ms] is input variable PT × high-speed timer setting value [ms] in the timer limit setting<sup>\*1</sup>.

\*1 The value set in [Navigation window] ⇒ [Parameter] ⇒ CPU module ⇒ [CPU Parameter] ⇒ [Operation Related Setting] ⇒ [Timer Limit Setting] ⇒ [High Speed Timer/High Speed Retentive Timer]

The valid setting range for the OFF delay timer time [ms] is determined by the long timer setting value in the timer limit setting<sup>\*2</sup>.

However, since the data type of the OFF delay timer time is TIME data type (32-bit value), the maximum value will fall within the range of TIME data type.

\*2 The value set in [Navigation window] ⇒ [Parameter] ⇒ CPU module ⇒ [CPU Parameter] ⇒ [Operation Related Setting] ⇒ [Timer Limit Setting] ⇒ [Long Timer/Long Retentive Timer]

- Minimum value

Identical to the long timer setting value [ms] in the timer limit setting.

Note that if the long timer setting value is smaller than 1ms, the minimum value will be 1ms.

- Maximum value

OFF delay timer time [ms] ≤ 2147483647 [ms] × long timer setting value [ms] in the timer limit setting

**Ex.**

If the long timer setting value is 0.001ms: T#1ms to T#2147400ms

If the long timer setting value is 1000ms: T#1000ms to T#32767000ms

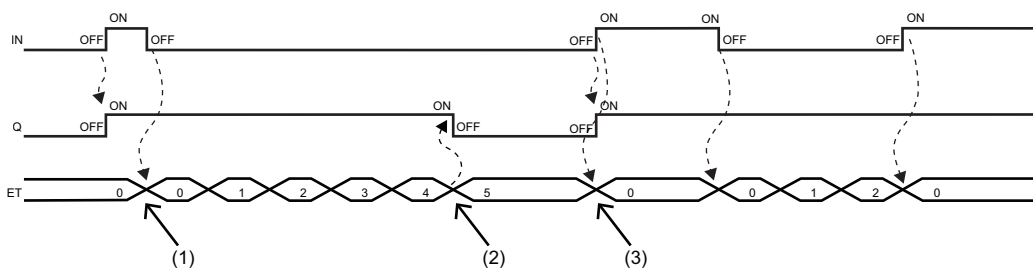
The value when the output variable Q turns on and off (falling edge) is used for the setting value of the input variable PT.

When the value of the input variable PT is changed while the input variable IN is off, the new value will be enabled at the next falling edge timing.

### ■ Operation result

Operation result	Q, ET
No operation error	Operation result output value
Operation error	Undefined value

When PT = 500 (when the high-speed timer setting value in the timer limit setting is 10ms) (5 seconds)



- (1) When IN = off, ET starts measuring time.
- (2) When the time measured in ET reaches the time set in PT, Q is turned off.
- (3) When IN turns on, ET is reset.

## Operation error

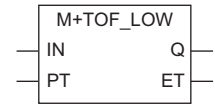
Error code (SD0)	Description
3401H	The OFF delay timer time [ms] exceeds the valid range.

# 13.6 OFF Delay Timer (Low-speed Timer Type) (M+TOF\_LOW)

## M+TOF\_LOW

This function block turns off a signal after the specified period of time.

### FBD/LD



### Point

A project using this FB cannot be opened with a version earlier than 1.080J of the engineering tool.

13

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
IN	Rising Edge Detection	TRUE, FALSE	Input variable	BOOL
PT	OFF Delay Timer Setting	1 to 32767	Input variable	INT
Q	Output	TRUE, FALSE	Output variable	BOOL
ET	Measuring Current Value	0 to 32767	Output variable	INT

## Processing details

### ■ Operation processing

[Output]

When the input variable IN turns on, this function turns on the output variable Q.

When the input variable IN turns on and off, this function block turns off the output variable Q after the period of time set in the input variable PT.

The time elapsed until the output variable Q turns off is set to the output variable ET.

Use the long timer to count the elapsed time.

[OFF delay timer setting]

The OFF delay timer time [ms] is input variable PT × low-speed timer setting value [ms] in the timer limit setting<sup>\*1</sup>.

\*1 The value set in [Navigation window] ⇒ [Parameter] ⇒ CPU module ⇒ [CPU Parameter] ⇒ [Operation Related Setting] ⇒ [Timer Limit Setting] ⇒ [Low Speed Timer/Low Speed Retentive Timer]

The valid setting range for the OFF delay timer time [ms] is determined by the long timer setting value in the timer limit setting<sup>\*2</sup>.

However, since the data type of the OFF delay timer time is TIME data type (32-bit value), the maximum value will fall within the range of TIME data type.

\*2 The value set in [Navigation window] ⇒ [Parameter] ⇒ CPU module ⇒ [CPU Parameter] ⇒ [Operation Related Setting] ⇒ [Timer Limit Setting] ⇒ [Long Timer/Long Retentive Timer]

- Minimum value

Identical to the long timer setting value [ms] in the timer limit setting.

Note that if the long timer setting value is smaller than 1ms, the minimum value will be 1ms.

- Maximum value

OFF delay timer time [ms] ≤ 2147483647 [ms] × long timer setting value [ms] in the timer limit setting

**Ex.**

If the long timer setting value is 0.001ms: T#1ms to T#2147400ms

If the long timer setting value is 1000ms: T#1000ms to T#32767000ms

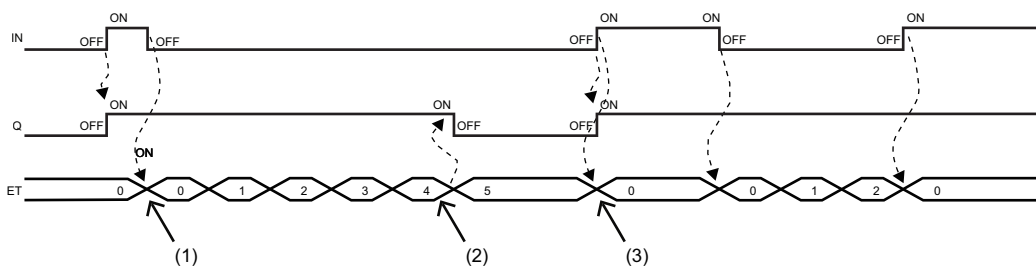
The value when the output variable Q turns on and off (falling edge) is used for the setting value of the input variable PT.

When the value of the input variable PT is changed while the input variable IN is off, the new value will be enabled at the next falling edge of IN.

### ■ Operation result

Operation result	Q, ET
No operation error	Operation result output value
Operation error	Undefined value

When PT = 500 (when the low-speed timer setting value in the timer limit setting is 10ms) (5 seconds)



- (1) When IN = off, ET starts measuring time.
- (2) When the time measured in ET reaches the time set in PT, Q is turned off.
- (3) When IN turns on, ET is reset.

## Operation error

Error code (SD0)	Description
3401H	The OFF delay timer time [ms] exceeds the valid range.



This part consists of the following chapters.

14 I/O CONTROL

---

15 LOOP CONTROL OPERATION

---

16 TAG SPECIAL

---

# 14 I/O CONTROL

The following FBs perform I/O processing including analog input/output, pulse integration, and batch counter.

**Point**

To set initial values of public variables of the tag access FB arranged on a user-defined tag FB in "FB Property" of the engineering tool, refer to the following.

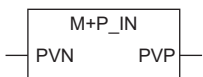
GX Works3 Operating Manual

## 14.1 Analog Input Processing (M+P\_IN)

### M+P\_IN

Performs range check, input limiter, engineering value inverse conversion, and digital filter processing.

FBD/LD



#### Applicable tag type

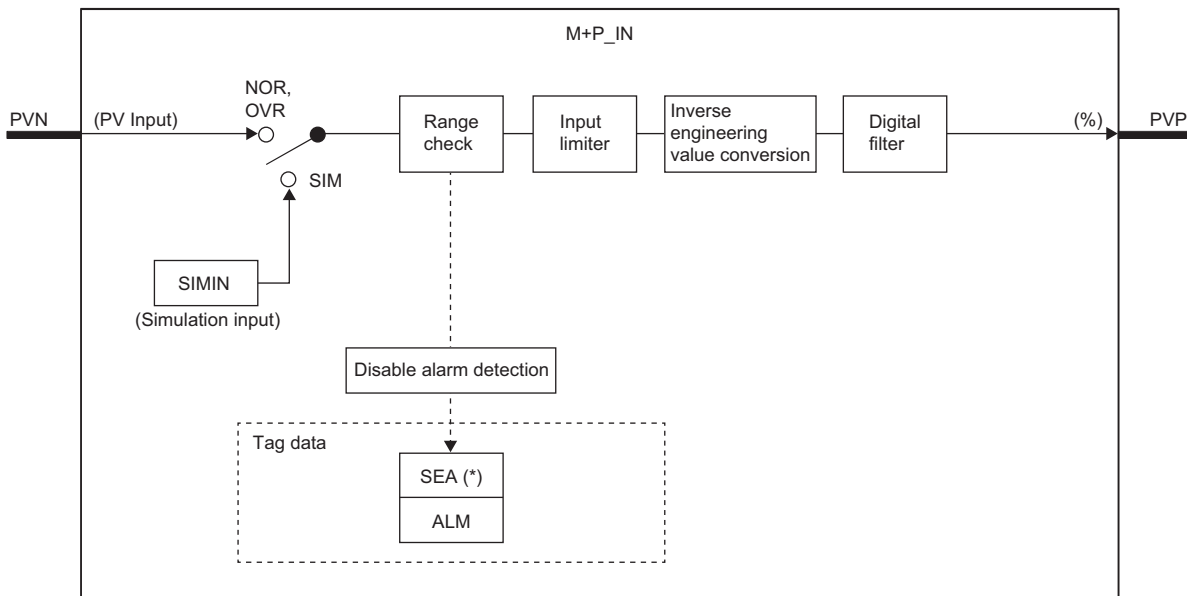
PID, 2PID, 2PIDH, PIDP, SPI, IPD, BPI, R, ONF2, ONF3, MONI, SWM, MWM, PVAL

#### Control mode

MAN	AUT	CAS*1	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

\*1 Transition to CASDR is possible.

#### Block diagram



\* Indicates bits item.

## Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	NMIN to NMAX	Input variable	REAL
PVP	PV output	0 to 100[%]	Output variable	REAL

### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
H_	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
L_	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
LMT_TYPE*2	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
SEA_OTYPE*1	Hold processing selection when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT

\*1 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

\*2 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

### ■Public variable (others) \*1


- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation input	NMIN to NMAX	0.0	User	REAL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

### ■Tag data

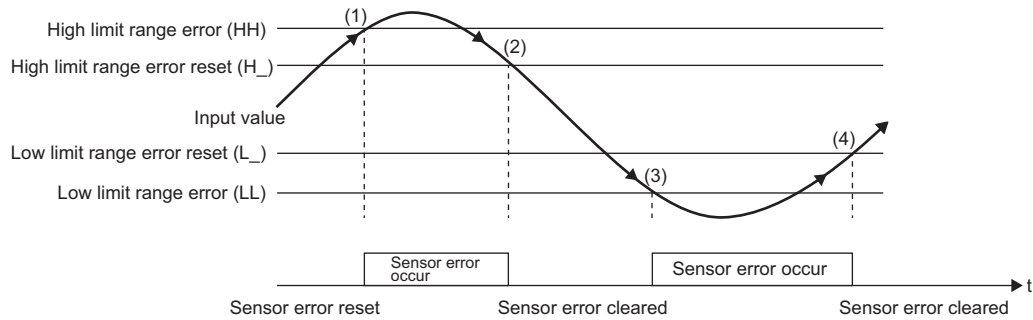
For details on tag data that is read or written with this tag access FB, refer to the following.

 Page 812 Tag Data List

## Processing details

### Range check

This function block checks the range of an input value.



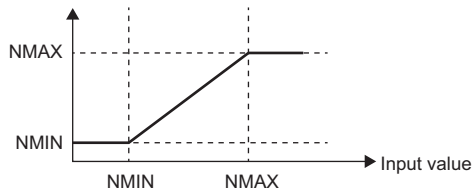
No.	Condition	Alarm (ALM)
		Sensor error (SEA)
(1)	Input value $\geq$ HH	TRUE (Detected)
(2)	Input value $\leq$ H_	FALSE (Reset)
(3)	Input value $\leq$ LL	TRUE (Detected)
(4)	Input value $\geq$ L_	FALSE (Reset)

HH: High limit range error  
H\_: High limit range error reset  
LL: Low limit range error  
L\_: Low limit range error reset

### Input limiter

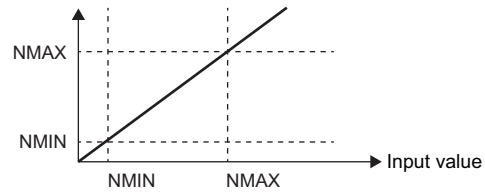
This function block performs input limiter processing.

Input limiter processing result



Operation when input limiter processing is enabled.

Input limiter processing result



Operation when input limiter processing is disabled.

Condition	Input limiter processing result
Input value $\geq$ NMAX	NMAX
Input value $\leq$ NMIN	NMIN
NMIN < Input value < NMAX	Input value

NMAX: Input high limit, NMIN: Input low limit

Input limiter processing is performed according to the set value in the input limiter processing selection (LMT\_TYPE).

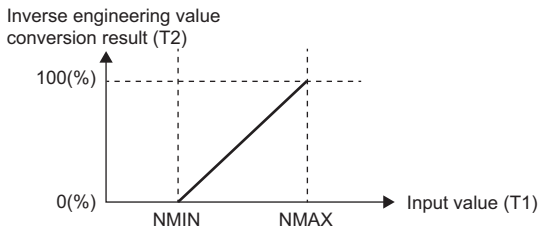
Condition	Processing result
LMT_TYPE = 0	Comply with the setting common to all tags.*1 • "Yes" is selected for "Process the Input Limiter of M+P_IN"; Input limiter processing is performed. • "No" is selected for "Process the Input Limiter of M+P_IN"; Input limiter processing is not performed.
LMT_TYPE = 1	Set the individual operation for each tag. Input limiter processing is performed.
LMT_TYPE = 2	Set the individual operation for each tag. Input limiter processing is not performed.

\*1 The setting common to all tags is set in the process control extended setting in the engineering tool.

[Options] ⇒ [Convert] ⇒ [Process Control Extension Setting] ⇒ [I/O Control] ⇒ [Input Limiter]

### ■Engineering value inverse conversion

This function block converts an engineering value input from the A/D converter module into a value in percentage (%).



$$T2 (\%) = \frac{T1 - NMIN}{NMAX - NMIN} \times 100 (\%)$$

- T1: Input value
- T2: Engineering value inverse conversion processing result (%)
- NMAX: Input high limit
- NMIN: Input low limit

### ■Digital filter

This function block performs digital filter processing to an input value.

Digital filter processing result =  $T2 + \alpha$  (Last digital filter processing value - T2)

$\alpha$ : Filter coefficient, T2: Engineering value inverse conversion processing result

### ■Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the range check.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no SEA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• SEI</li> </ul>
Disable alarm detection by loop stop processing	Page 187 Loop stop processing

### ■Hold processing

This function block sets whether or not to hold M+P\_IN outputs according to SEA\_OTYPE when a sensor error (SEA) caused by a high/low limit range error occurs in the range check.\*1

Condition	Processing result
SEA_OTYPE(0)	Comply with the setting common to all tags.*2 <ul style="list-style-type: none"> <li>• "Yes" is selected for "Hold the output of M+P_IN"; Outputs are held.</li> <li>• "No" is selected for "Hold the output of M+P_IN"; The operation continues.</li> </ul>
SEA_OTYPE(1)	Set the individual operation for each tag. Outputs are held.
SEA_OTYPE(2)	Set the individual operation for each tag. The operation continues.

\*1 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*2 The setting common to all tags is set in the process control extended setting in the engineering tool.

[Options] ⇒ [Convert] ⇒ [Process Control Extension Setting] ⇒ [I/O Control] ⇒ [Holding Processing]

### ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) or the tag stop (TSTP) of the monitor output buffer (DOM) is TRUE, this function block performs the following processing.

- Outputs (PVP) are held.
- The control mode is automatically switched to MANUAL.
- When SEA of the alarm (ALM) has occurred, the SEA is reset. When TSTP is TRUE, SPA is reset.
- Alarm detections are not performed in the range check.

## ■ Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation				
	Range check	Input limiter	Engineering value inverse conversion	Digital filter	Alarm
MAN, CMV, AUT, CAS, CSV, CASDR	○	○	○	○	○*1

\*1 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVN), block memory, operation constant (NMAX, NMIN, HH, H_, L_, LL) or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	High limit range error occurrence (HH) < high limit range error return (H_), low limit range error return (L_) < low limit range error occurrence (LL), or input high limit (NMAX) < input low limit (NMIN)

### Point

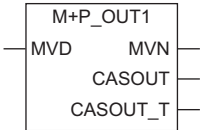
Initial values of the high/low range error and high/low range error reset are based on the default input range of the analog input module. The values are digital values converted into values in percentage. To change the input range, change the initial values as necessary.

# 14.2 Output Processing-1 with Mode Switching (With Input Addition) (M+P\_OUT1)

## M+P\_OUT1

This FB performs input addition, variation rate & high/low limiter, reset windup, and output conversion processing to an input value ( $\Delta MV$ ) and outputs a manipulated value. (with integral and anti-reset windup processing)

### FBD/LD



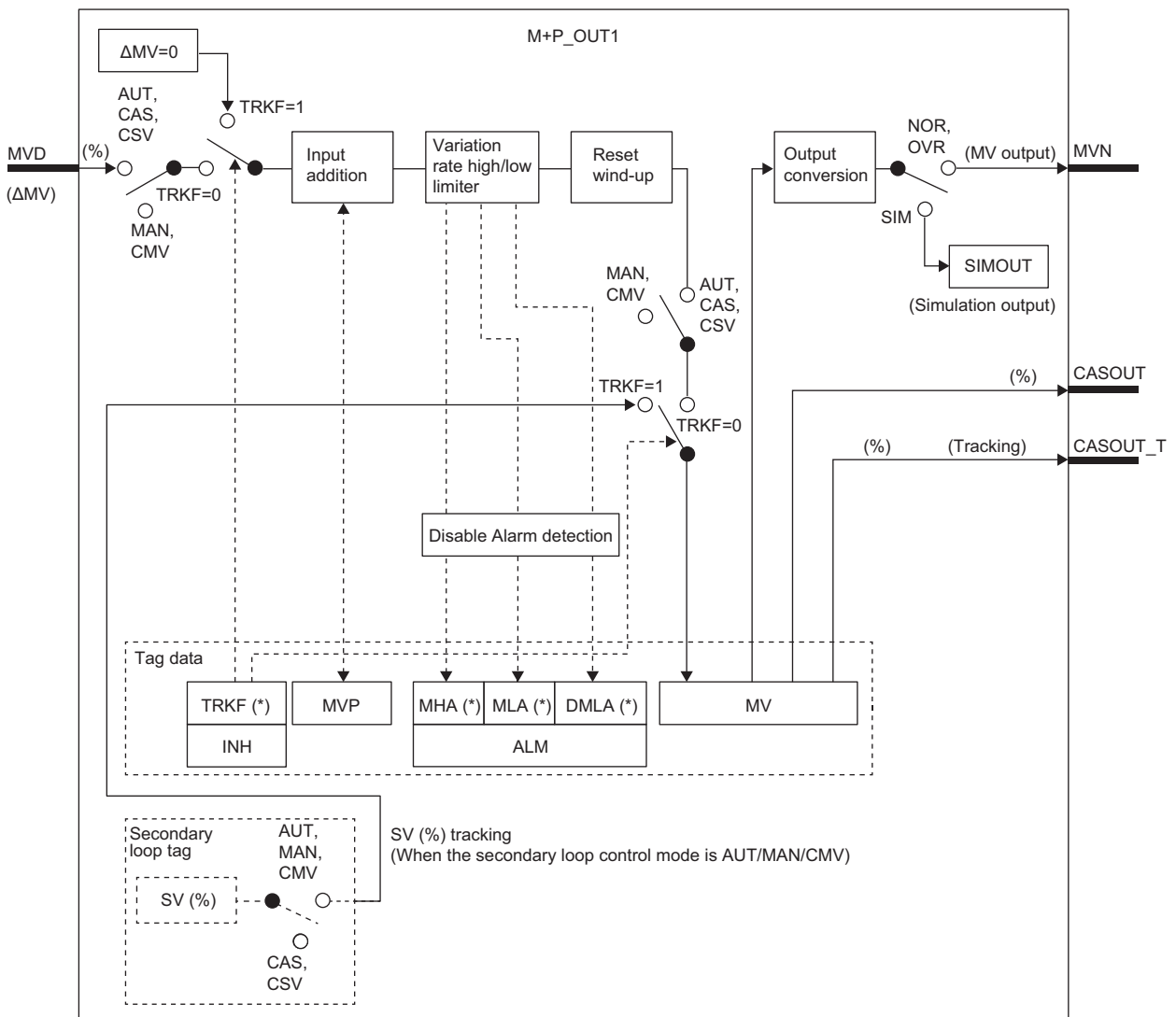
### Applicable tag type

BPI, IPD, PID, SPI, 2PID

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bits item.

## Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
MVD	ΔMV input	-999999 to 999999[%]	Input variable	REAL
MVN	Output to a module	NMIN to NMAX	Output variable	REAL
CASOUT	Cascade output	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade output (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
SEA_OTYPE <sup>*1</sup>	Hold processing selection when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT

\*1 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

### ■Public variable (others) <sup>\*1</sup>


- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

### ■Tag data

For details on tag data that is read or written with this tag access FB, refer to the following.

 Page 812 Tag Data List

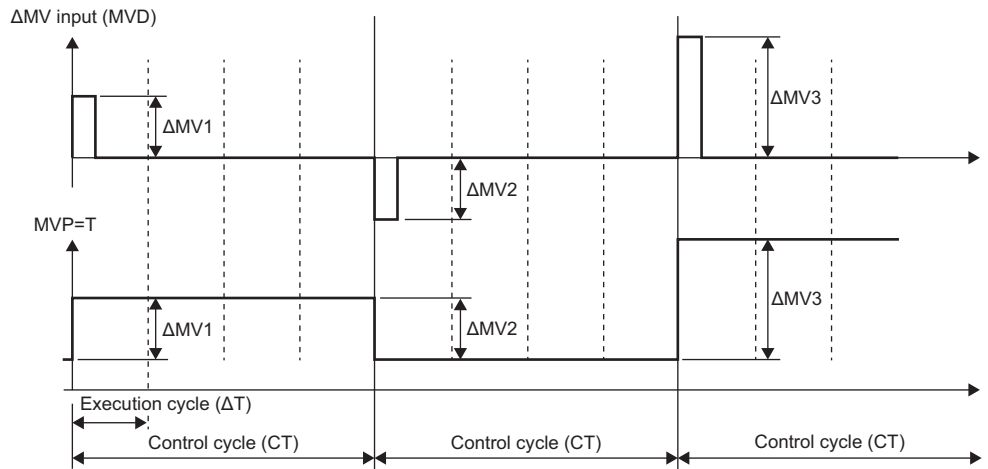


## Processing details

### Input addition

A tentative manipulated value (T) is calculated from the input value ( $\Delta MV$ ).

( $\Delta MV$  is output from an FB (such as M+P\_PID) before this FB every control cycle (CT).)



$$T = \Delta MV + MVP$$

$$MVP = T$$

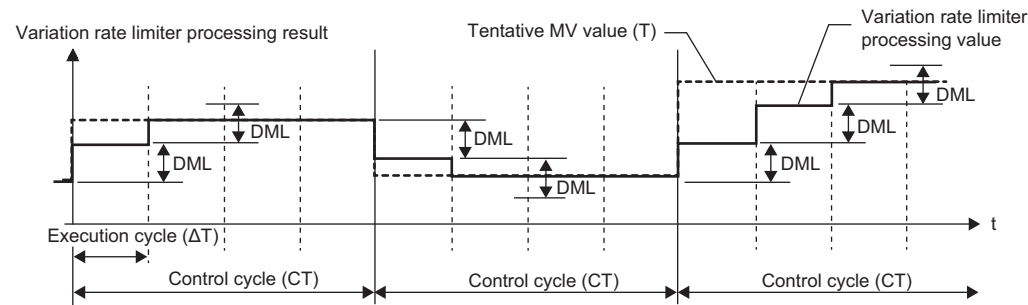
$\Delta MV$ : Input value to input (MVD), MVP: MV internal operation value, T: Tentative manipulated value

- When the control mode is switched from MAN or CMV to AUT or CSV, a manipulated value (MV) is stored in the MV internal operation value (MVP) to avoid sudden changes of the manipulated value at switching.
- During data tacking,  $\Delta MV$  is 0 and the manipulated value (MV) is stored in the MV internal operation value (MVP).

### Variation rate & high/low limiter

This function block checks the variation rate and high/low limits of the input value.

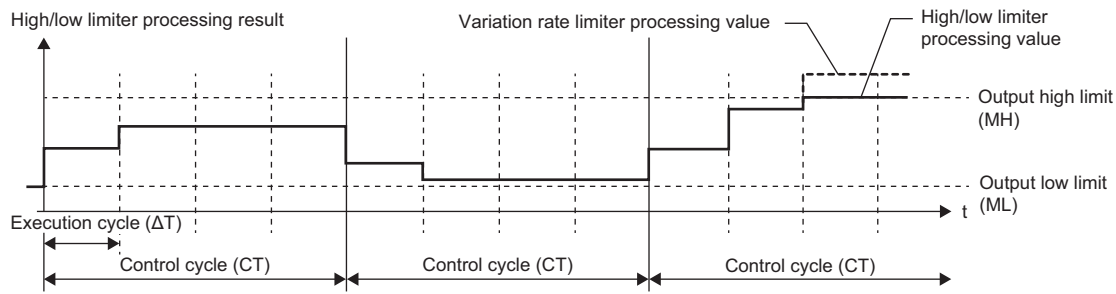
- Variation rate limiter



Condition	Variation rate limiter processing result	Alarm (ALM)
		Output variation rate limit (DMLA)
$ T - MV  \leq DML$	T	FALSE (Reset)
$T - MV > DML$	$MV + DML$	TRUE (Detected)
$T - MV < -DML$	$MV - DML$	TRUE (Detected)

T: Tentative manipulated value, MV: Manipulated value, DML: Output variation rate high limit value

• High/low limiter



Condition	High/low limiter processing result	Alarm (ALM)	
		Output low limit (MLA)	Output high limit (MHA)
Variation rate limiter processing result > MH	MH	FALSE (Reset)	TRUE (Detected)
Variation rate limiter processing result < ML	ML	TRUE (Detected)	FALSE (Reset)
$ML \leq$ Variation rate limiter processing result $\leq$ MH	Variation rate limiter processing value	FALSE (Reset)	FALSE (Reset)

MH: Output high limit value, ML: Output low limit value

■Reset windup

As countermeasures against reset windup, this function block sets a manipulated value to the high/low limit value when the manipulated value has exceeded the limit value, and quickly responds to a control target when deviation is inverted.

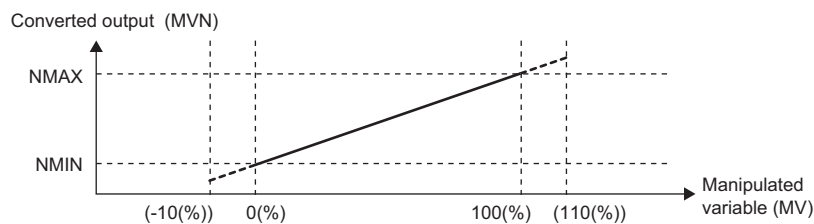
Condition	Countermeasure processing against reset windup
Variation rate limiter processing result > MH and $\frac{\Delta T}{Ti} \leq 1$	$MVP = \frac{\Delta T}{Ti} \times (MH - T) + T$
Variation rate limiter processing result < ML and $\frac{\Delta T}{Ti} \leq 1$	$MVP = \frac{\Delta T}{Ti} \times (ML - T) + T$

MH: Output high limit value  
 ML: Output low limit value  
 MVP: MV internal operation value  
 $\Delta T$ : Execution cycle  
 T: Tentative manipulated value  
 Ti: Integral time

However, when Ti is 0, countermeasure processing against reset windup is not performed.

■Output conversion

This function block performs output conversion processing.




$$\text{Converted output (MVN)} = \left\{ (NMAX - NMIN) \times \frac{MV}{100} \right\} + NMIN$$

NMAX: Output conversion high limit value  
 NMIN: Output conversion low limit value  
 MV: Manipulated variable (%)  
 MVN: Output conversion output value

### ■Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the variation rate & high/low limiter processing.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no DMLA, MHA, or MLA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DMLI</li> <li>• MHI</li> <li>• MLI</li> </ul>
Disable alarm detection by control mode selection	In MAN or CMV mode, DMLA, MHA, and MLA of the alarm (ALM) are reset and DMLA, MHA, or MLA will not be detected.
Disable alarm detection by loop stop processing	 Page 193 Loop stop processing

### ■Hold processing

This function block sets whether to hold M+P\_OUT1 outputs or not according to the set value in the hold processing selection (SEA\_OTYPE) when a sensor error (SEA) has occurred in M+P\_IN, a tag access FB.

Condition	Processing result
SEA_OTYPE = 0	Comply with the setting common to all tags.*1 <ul style="list-style-type: none"> <li>• "Yes" is selected for "Hold the output of M+P_OUT1, M+P_OUT2, M+P_DUTY"; Outputs are held.</li> <li>• "No" is selected for "Hold the output of M+P_OUT1, M+P_OUT2, M+P_DUTY"; The operation continues.</li> </ul>
SEA_OTYPE = 1	Set the individual operation for each tag. Outputs are held.
SEA_OTYPE = 2	Set the individual operation for each tag. The operation continues.

\*1 The setting common to all tags is set in the process control extended setting in the engineering tool.

 [Options] ⇒ [Convert] ⇒ [Process Control Extension Setting] ⇒ [I/O Control] ⇒ [Holding Processing]

### ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- Outputs (MVN) are held.
- The control mode is automatically switched to MANUAL.
- When DMLA, MHA, or MLA of the alarm (ALM) has occurred, the DMLA, MHA, or MLA is reset.
- No alarm is detected in the variation rate & high/low limiter processing.

### ■Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation				
	Input addition	Variation rate & high/low limiter	Reset windup	Output conversion	Alarm
MAN, CMV	×	×	×	○	×*1
AUT, CAS, CSV	○	○	○	○	○*2

\*1 An alarm (ALM) whose corresponding bit is TRUE (detected) is reset, and the alarm will not be detected.

\*2 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

### Operation error

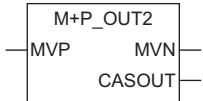
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (MVD), operation constant (NMAX, NMIN) or tag data memory is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# 14.3 Output Processing-2 with Mode Switching (Without Input Addition) (M+P\_OUT2)

## M+P\_OUT2

This FB performs variation rate & high/low limiter processing and output conversion processing to an input value (MV), and outputs a manipulated value. (without integral and anti-reset windup processing)

### FBD/LD



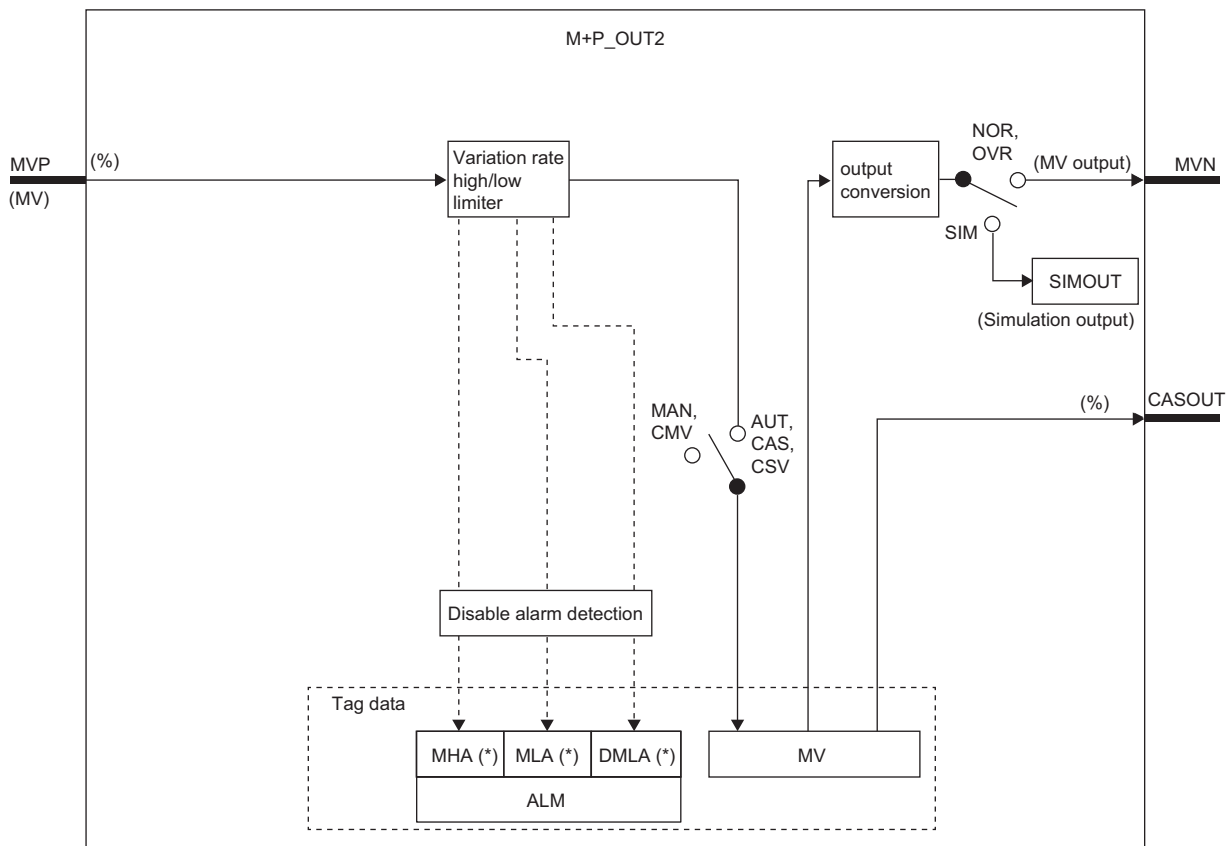
### Applicable tag type

R

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bits item.

## Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
MVP	MV input	0 to 100[%]	Input variable	REAL
MVN	Output to a module	NMIN to NMAX	Output variable	REAL
CASOUT	Cascade output	0 to 100[%]	Output variable	REAL

### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
SEA_OTYPE <sup>*1</sup>	Hold processing selection when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT

\*1 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

### ■Public variable (others) <sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

### ■Tag data

For details on tag data that is read or written with this tag access FB, refer to the following.

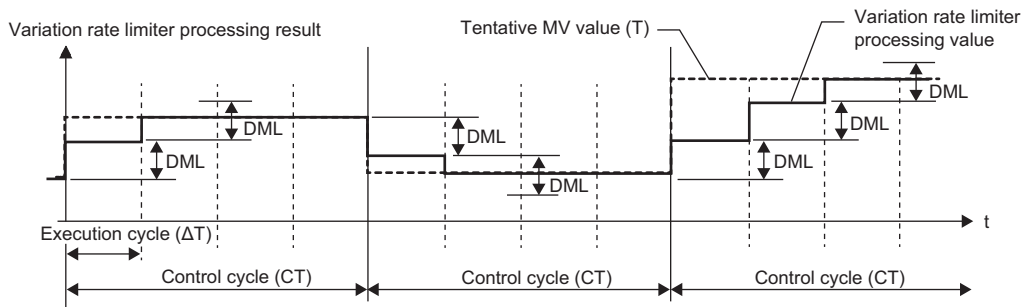
☞ Page 845 R

## Processing details

### ■ Variation rate & high/low limiter

This function block checks the variation rate and high/low limits of the input value.

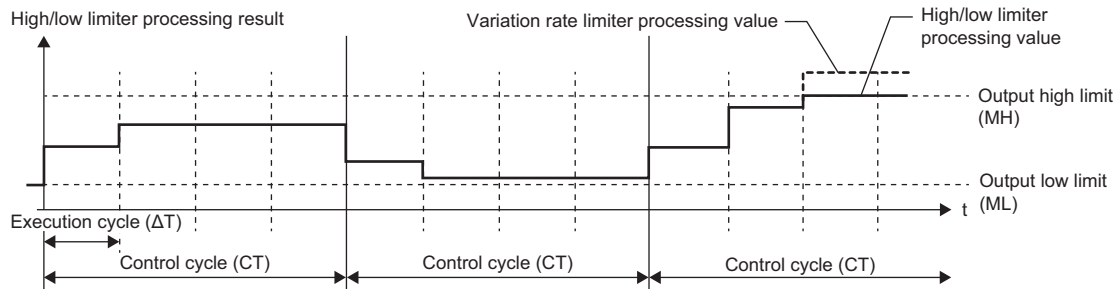
- Variation rate limiter



Condition	Variation rate limiter processing result	Alarm (ALM)
		Output variation rate limit (DMLA)
$ T - MV  \leq DML$	T	FALSE (Reset)
$T - MV > DML$	$MV + DML$	TRUE (Detected)
$T - MV < -DML$	$MV - DML$	TRUE (Detected)

T: Tentative manipulated value, MV: Manipulated value, DML: Output variation rate high limit value

- High/low limiter

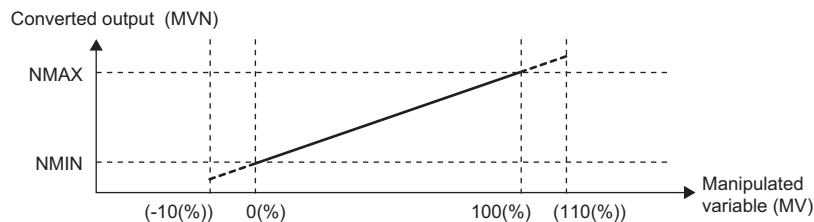


Condition	High/low limiter processing result	Alarm (ALM)	
		Output low limit (MLA)	Output high limit (MHA)
Variation rate limiter processing result $> MH$	MH	FALSE (Reset)	TRUE (Detected)
Variation rate limiter processing result $< ML$	ML	TRUE (Detected)	FALSE (Reset)
$ML \leq$ Variation rate limiter processing result $\leq MH$	Variation rate limiter processing value	FALSE (Reset)	FALSE (Reset)

MH: Output high limit value, ML: Output low limit value

### ■ Output conversion

This function block performs output conversion processing.



$$\text{Converted output (MVN)} = \left\{ (NMAX - NMIN) \times \frac{MV}{100} \right\} + NMIN$$

NMAX: Output conversion high limit value

NMIN: Output conversion low limit value

MV: Manipulated variable (%)

MVN: Output conversion output value

### ■Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the variation rate & high/low limiter processing.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no DMLA, MHA, or MLA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DMLI</li> <li>• MHI</li> <li>• MLI</li> </ul>
Disable alarm detection by control mode selection	In MAN or CMV mode, DMLA, MHA, and MLA of the alarm (ALM) are reset and DMLA, MHA, or MLA will not be detected.
Disable alarm detection by loop stop processing	Page 197 Loop stop processing

### ■Hold processing

This function block sets whether to hold M+P\_OUT2 outputs or not according to the set value in the hold processing selection (SEA\_OTYPE) when a sensor error (SEA) has occurred in M+P\_IN, a tag access FB.

Condition	Processing result
SEA_OTYPE = 0	Comply with the setting common to all tags.*1 <ul style="list-style-type: none"> <li>• "Yes" is selected for "Hold the output of M+P_OUT1, M+P_OUT2, M+P_DUTY"; Outputs are held.</li> <li>• "No" is selected for "Hold the output of M+P_OUT1, M+P_OUT2, M+P_DUTY"; The operation continues.</li> </ul>
SEA_OTYPE = 1	Set the individual operation for each tag. Outputs are held.
SEA_OTYPE = 2	Set the individual operation for each tag. The operation continues.

\*1 The setting common to all tags is set in the process control extended setting in the engineering tool.

[Options] ⇒ [Convert] ⇒ [Process Control Extension Setting] ⇒ [I/O Control] ⇒ [Holding Processing]

### ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- Outputs (MVN) are held.
- The control mode is automatically switched to MANUAL.
- When DMLA, MHA, or MLA of the alarm (ALM) has occurred, the DMLA, MHA, or MLA is reset.
- No alarm is detected in the variation rate & high/low limiter processing.

### ■Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation		
	Variation rate & high/low limiter	Output conversion	Alarm
MAN, CMV	×	○	×*1
AUT, CAS, CSV	○	○	○*2

\*1 An alarm (ALM) whose corresponding bit is TRUE (detected) is reset, and the alarm will not be detected.

\*2 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

### Operation error

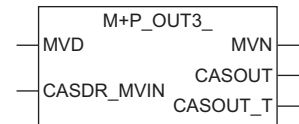
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (MVP) or operation constant (NMAX, NMIN), or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# 14.4 Output Processing-3 with Mode Switching (With Input Addition and Compensation) (M+P\_OUT3\_)

## M+P\_OUT3\_

This FB performs input addition, MV compensation, preset MV, MV hold, MV tracking, variation rate & high/low limiter, reset windup, tight shut/full open, MV reverse, and output conversion processing to an input value ( $\Delta MV$ ) and outputs a manipulated value.

### FBD/LD



### Applicable tag type

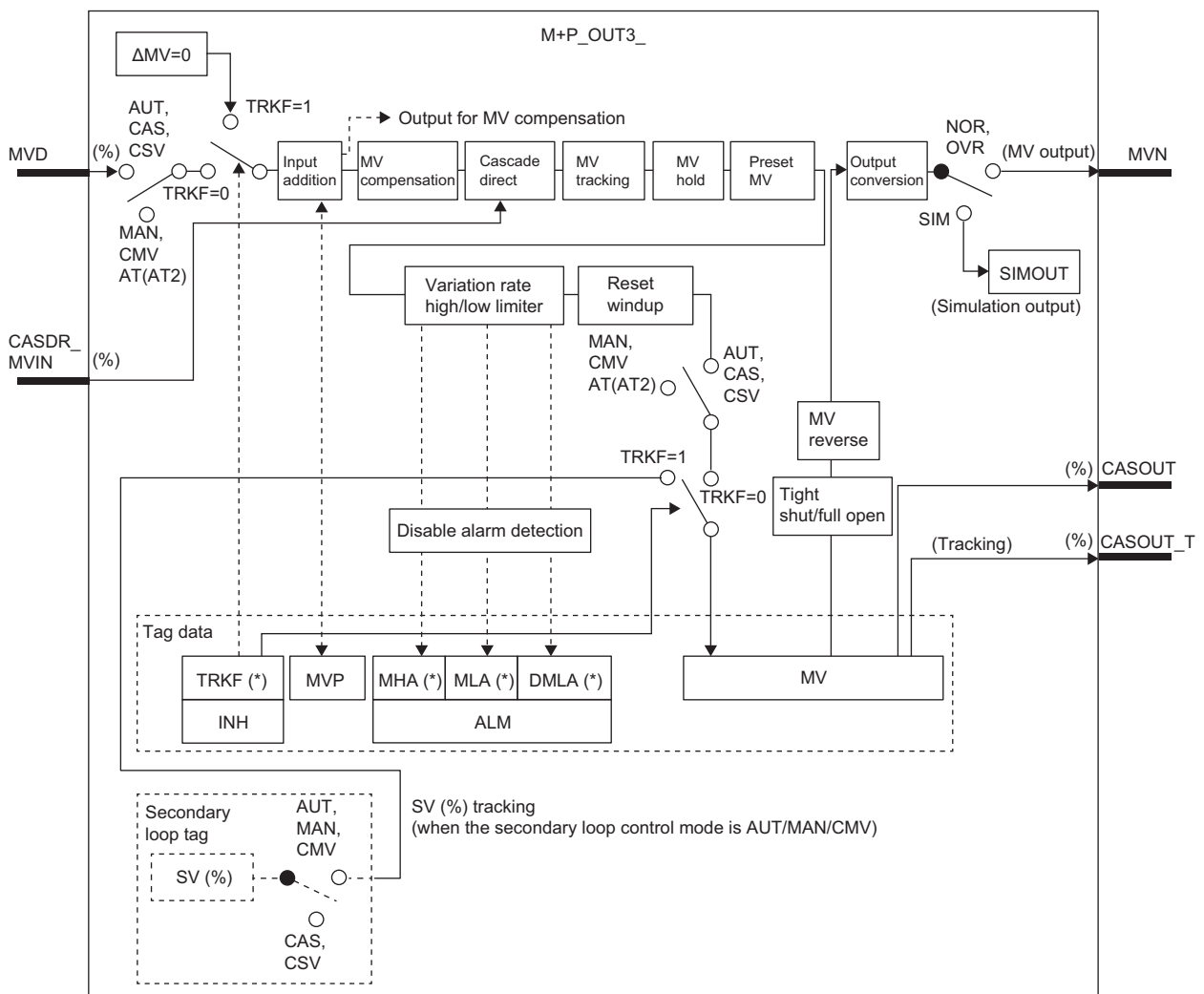
2PIDH

### Control mode

MAN	AUT	CAS*1	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

\*1 Transition to CASDR is possible.

### Block diagram



\* Indicates bits item.



## Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
MVD	ΔMV input	-999999 to 999999[%]	Input variable	REAL
CASDR_MVIN	MV input for cascade direct	0 to 100[%]	Input variable	REAL
MVN	Output to a module	NMIN to NMAX	Output variable	REAL
CASOUT	Cascade output	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade output (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
MVCMP_EN	MV Compensation Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
MV_CMPIN	MV compensation value	-999999 to 999999[%]	0.0	User	REAL
MVCMP_MODE	MV Compensation Mode	0: Addition 1: Replacement	0	User	INT
PREMV_EN	Preset MV Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
PREMV_V	Preset manipulated value	0 to 100[%]	0.0	User	REAL
MVHLD_EN	MV hold execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
MVTRK_EN	MV Tracking Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
MV_TRKIN	MV tracking input	0 to 100[%]	0.0	User	REAL
STP_OTYPE	Output when loop or tag is stopped	0: Hold 1: Preset value	0	User	INT
SEA_OTYPE	MV output selection at occurrence of SEA	0: Hold 1: Preset MV output 2: Neither hold nor preset MV output is performed.	0	User	INT
ARW_EX_EN	MV Value Instantaneous Pullback	TRUE: Execute FALSE: Not execute	FALSE	User	BOOL
MVPH	MV internal operation value high limit	MH to 999999[%]	100.0	User	REAL
MVPL	MV internal operation value low limit	-999999 to ML[%]	0.0	User	REAL
MVREV_EN	MV Reverse Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
FOTS_EN	Tight Shut/Full Open Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
MVFO	Output value for full open	100 to 125[%]	112.5	User	REAL
MVTS	Output value for tight shut	-25 to 0[%]	-16.82	User	REAL

## ■Public variable (others) \*1

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

- MV compensation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
MV_CMPOUT	Output for MV compensation	-999999 to 999999[%]	0.0	System	REAL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■Tag data

For details on tag data that is read or written with this tag access FB, refer to the following.

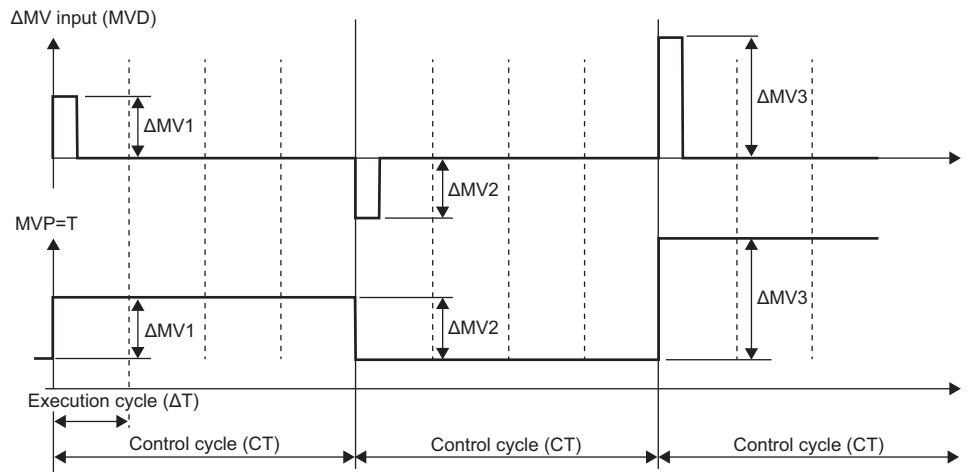
☞ Page 823 2PIDH

## Processing details

### ■Input addition

A tentative manipulated value (T) is calculated from the input value ( $\Delta MV$ ).

( $\Delta MV$  is output from an FB (such as M+P\_PID) before this FB every control cycle (CT).)



$$T = \Delta MV + MVP$$

$$MVP = T$$

$\Delta MV$ : Input value to input (MVD), MVP: MV internal operation value, T: Tentative manipulated value

- When the control mode is switched from MAN or CMV to AUT or CSV, a manipulated value (MV) is stored in the MV internal operation value (MVP) to avoid sudden changes of the manipulated value at switching.
- During data tacking,  $\Delta MV$  is 0 and the manipulated value (MV) is stored in the MV internal operation value (MVP).

### ■MV compensation

This function block performs compensation processing to the tentative manipulated value.

Condition		Processing result
MVCMP_EN = TRUE	MVCMP_MODE = 0 (addition)	T + MV_CMPIN
	MVCMP_MODE = 1 (replacement)	MV_CMPIN
MVCMP_EN = FALSE	MVCMP_MODE = 0 (addition)	T
	MVCMP_MODE = 1 (replacement)	T

MVCMP\_EN: MV compensation execution condition

T: Tentative manipulated value

MV\_CMPIN: MV compensation value

MVCMP\_MODE: MV compensation mode

### ■ Cascade direct

This function block sets the tentative manipulated value as a manipulated value of the primary loop.

Condition	Processing result
Control mode = CASCADE DIRECT (CAS = TRUE and CASDR = TRUE)	CASDR_MVIN
Control mode ≠ CASCADE DIRECT (CASDR = FALSE)	T

CASDR\_MVIN: MV input for cascade direct, T: Tentative manipulated value

### ■ MV tracking

This function block switches a manipulated value with a tracking input.

Condition	Processing result
MVTRK_EN = TRUE	MV_TRKIN
MVTRK_EN = FALSE	T

MVTRK\_EN: MV tracking execution condition, T: Tentative manipulated value, MV\_TRKIN: Tracking input

### ■ MV hold

This function block holds a manipulated value at that time.

Condition	Processing result
MVHLD_EN = TRUE	MVn-1
MVHLD_EN = FALSE	T

MVHLD\_EN: MV hold execution condition, T: Tentative manipulated value, MVn-1: Last manipulated value

### ■ Preset MV

This function block switches a tentative manipulated value with the preset manipulated value.

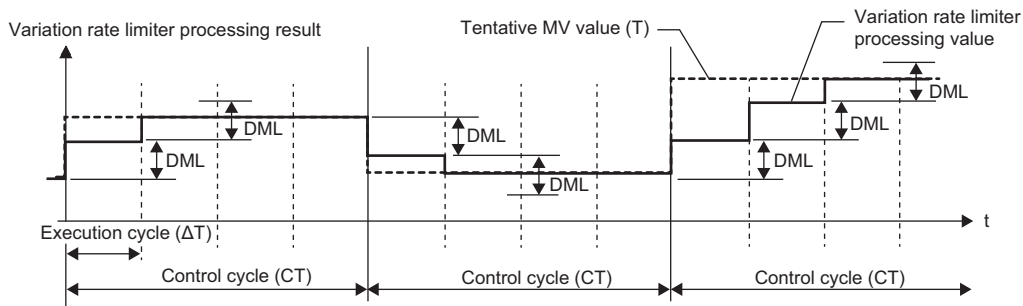
Condition	Processing result
PREMV_EN = TRUE	PREMV_V
PREMV_EN = FALSE	T

PREMV\_EN: Preset MV execution condition, T: Tentative manipulated value, PREMV\_V: Preset manipulated value

## Variation rate & high/low limiter

This function block checks the variation rate and high/low limits of the input value.

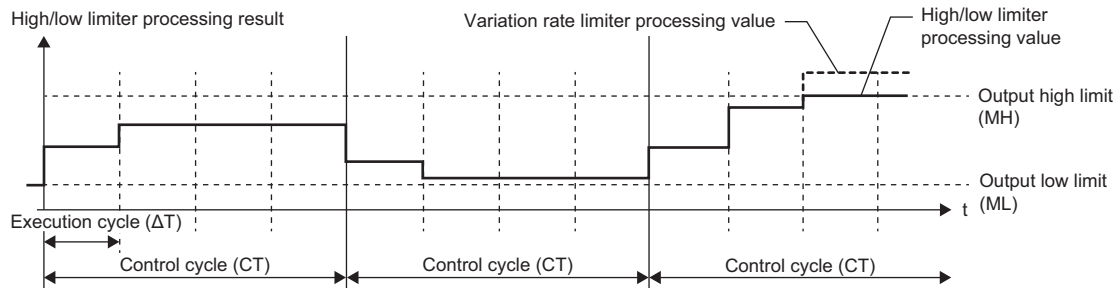
- Variation rate limiter



Condition	Variation rate limiter processing result	Alarm (ALM)
		Output variation rate limit (DMLA)
$ T - MV  \leq DML$	T	FALSE (Reset)
$T - MV > DML$	$MV + DML$	TRUE (Detected)
$T - MV < -DML$	$MV - DML$	TRUE (Detected)

T: Tentative manipulated value, MV: Manipulated value, DML: Output variation rate high limit value

- High/low limiter



Condition	High/low limiter processing result	Alarm (ALM)	
		Output low limit (MLA)	Output high limit (MHA)
Variation rate limiter processing result > MH	MH	FALSE (Reset)	TRUE (Detected)
Variation rate limiter processing result < ML	ML	TRUE (Detected)	FALSE (Reset)
$ML \leq \text{Variation rate limiter processing result} \leq MH$	Variation rate limiter processing value	TRUE (Detected)	FALSE (Reset)

MH: Output high limit value, ML: Output low limit value

## Reset windup

This function block performs processing without considering the value of ARW\_EX\_EN.

As countermeasures against reset windup, this function block sets a manipulated value to the high/low limit value when the manipulated value has exceeded the limit value, and quickly responds to a control target when deviation is inverted.

Condition	Countermeasure processing against reset windup
$MHA=1$ and $\frac{\Delta T}{T_i} \leq 1$	$MVP = \frac{\Delta T}{T_i} \times (MH - T) + T$
$MLA=1$ and $\frac{\Delta T}{T_i} \leq 1$	$MVP = \frac{\Delta T}{T_i} \times (ML - T) + T$

MHA: Output high limit alarm  
 MLA: Output low limit alarm  
 MH: Output high limit value  
 ML: Output low limit value  
 MVP: MV internal operation value  
 $\Delta T$ : Execution cycle  
 T: Tentative manipulated value  
 $T_i$ : Integral time

However, when  $T_i$  is 0, countermeasure processing against reset windup is not performed.

- When ARW\_EX\_EN is TRUE

As countermeasures against reset windup, this function block immediately sets a manipulated value to the MV internal operation high/low limit value when the manipulated value has exceeded the limit value, and quickly responds to a control target when deviation is inverted.

Condition	Countermeasure processing against reset windup
MVP > MVPH	MVP = MVPH
MVP < MVPL	MVP = MVPL

MVP: MV internal operation value, MVPH: MV internal operation value high limit, MVPL: MV internal operation value low limit

However, when MVPH is smaller than MH, MH is used instead of MVPH in the condition and as the pullback value.

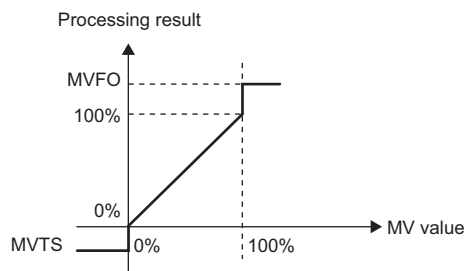
When MVPL is larger than ML, ML is used instead of MVPL in the condition and as the pullback value.

### ■Tight shut/full open

This tight shut/full open function is used to completely open or close a control valve.

The processing result is reduced to the output value for tight shut when a manipulated value is 0% or lower, and the result is increased to the output value for full open when a manipulated value is 100% or higher.

- When FOTS\_EN is TRUE



MVTS: Output value for tight shut (%), MVFO: Output value for full open (%)

### ■MV reverse

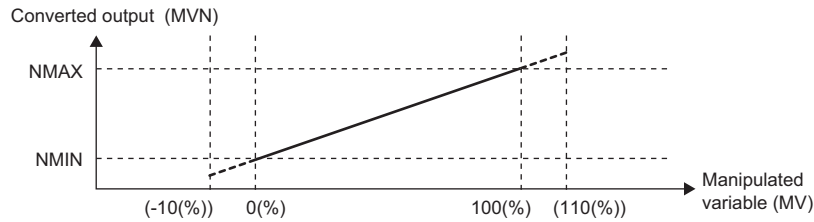
This function block performs inversion processing ( $100 - MV$ ) to a manipulated value.

Condition	Processing result
MVREV_EN = TRUE	MVREV = $100 - MV$
MVREV_EN = FALSE	MVREV = MV

MVREV: Output after processing of MV reverse for internal operation (%), MV: Manipulated value (%)

## ■Output conversion

This function block performs output conversion processing.



$$\text{Converted output (MVN)} = \left\{ (\text{NMAX} - \text{NMIN}) \times \frac{\text{MVREV}}{100} \right\} + \text{NMIN}$$

NMAX: Output conversion high limit value

NMIN: Output conversion low limit value

MVREV: Output after processing of MV reverse for internal operation (%)

MVN: Output conversion output value

## ■Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the variation rate & high/low limiter processing.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no DMLA, MHA, or MLA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERR1</li> <li>• DMLI</li> <li>• MHI</li> <li>• MLI</li> </ul>
Disable alarm detection by control mode selection	In MAN or CMV mode, DMLA, MHA, and MLA of the alarm (ALM) are reset and DMLA, MHA, or MLA will not be detected.
Disable alarm detection by loop stop processing	Page 204 Loop stop processing

## ■Output processing at occurrence of SEA

This function block selects one of the following three conditions as M+P\_OUT3\_ outputs when a sensor error (SEA) has occurred in M+P\_IN, a tag access FB.

Condition	Processing result
SEA_OTYPE = 0	MV hold
SEA_OTYPE = 1	Preset MV output
SEA_OTYPE = 2	Neither hold nor preset MV output is performed.

## ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) or the tag stop (TSTP) of the monitor output buffer (DOM) is TRUE, this function block performs the following processing.

- When the output at loop stop is the last value (STP\_OTYPE = 0), this function block holds the output (MVN). When the output at loop stop is the preset value (STP\_OTYPE = 1), this function block presets the output (MVN).
- The control mode is automatically switched to MANUAL.
- When DMLA, MHA, or MLA of the alarm (ALM) has occurred, the DMLA, MHA, or MLA is reset.
- No alarm is detected in the variation rate & high/low limiter processing.

## ■Auto tuning (AT2)

When auto tuning in the limit cycle method is being executed, this function block performs processing in the same way as the one in the MANUAL mode. ( Page 920 Limit cycle method)

### ■ Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation						
	Loop stop	Mode determination	Input addition	MV compensation	Cascade direct	Preset MV	MV hold
MAN, CMV	○	○	×	×	×	×	×
AUT, CAS, CSV	○	○	○	○	×	○	○
CASDR	○	○	○	×	○	○	○

Control mode	Processing operation						
	MV tracking	Variation rate & high/low limiter	Reset windup	MV reverse	Alarm	Auto tuning (AT2)	Output conversion
MAN, CMV	×	×	×	○	×*1	○	○
AUT, CAS, CSV	○	○	○	○	○*2	○	○
CASDR	○	○	○	○	○*2	×	○

\*1 An alarm (ALM) whose corresponding bit is TRUE (detected) is reset, and the alarm will not be detected.

\*2 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

### ■ Output function priority

The following table shows the priority of each output function.

Priority	Output function
1	Preset MV
2	MV hold
3	MV tracking
4	Cascade direct
5	MV compensation

**Ex.**

When both of the preset MV and MV tracking functions are valid (PREMV\_EN = TRUE, MVHLD\_EN = TRUE), a preset manipulated value is output because a higher priority is given to the preset MV function.

### Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Operation result of input data, operation constants, tag data, or inside of function blocks is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

**Point**

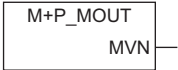
- The use of the tight shut/full open function is recommended in a combination with an isolated analog output module that has the range setting (extended mode) for allowing users to set a wide output range instead of the normal range (4 to 20mA, 1 to 5V).
- To use the tight shut/full open function with a module that does not have the extended mode in the range setting, set 0 to 20mA and 0 to 5V in the range setting and set the output conversion high/low limit values of this FB again.
- Output processing-3 with mode switching (with input addition and compensation) is performed every execution cycle (ΔT).

# 14.5 Manual Output (M+P\_MOUT)

## M+P\_MOUT

This FB reads a manipulated value (MV) of tag data, performs output conversion processing, and outputs a manipulated value.

### FBD/LD



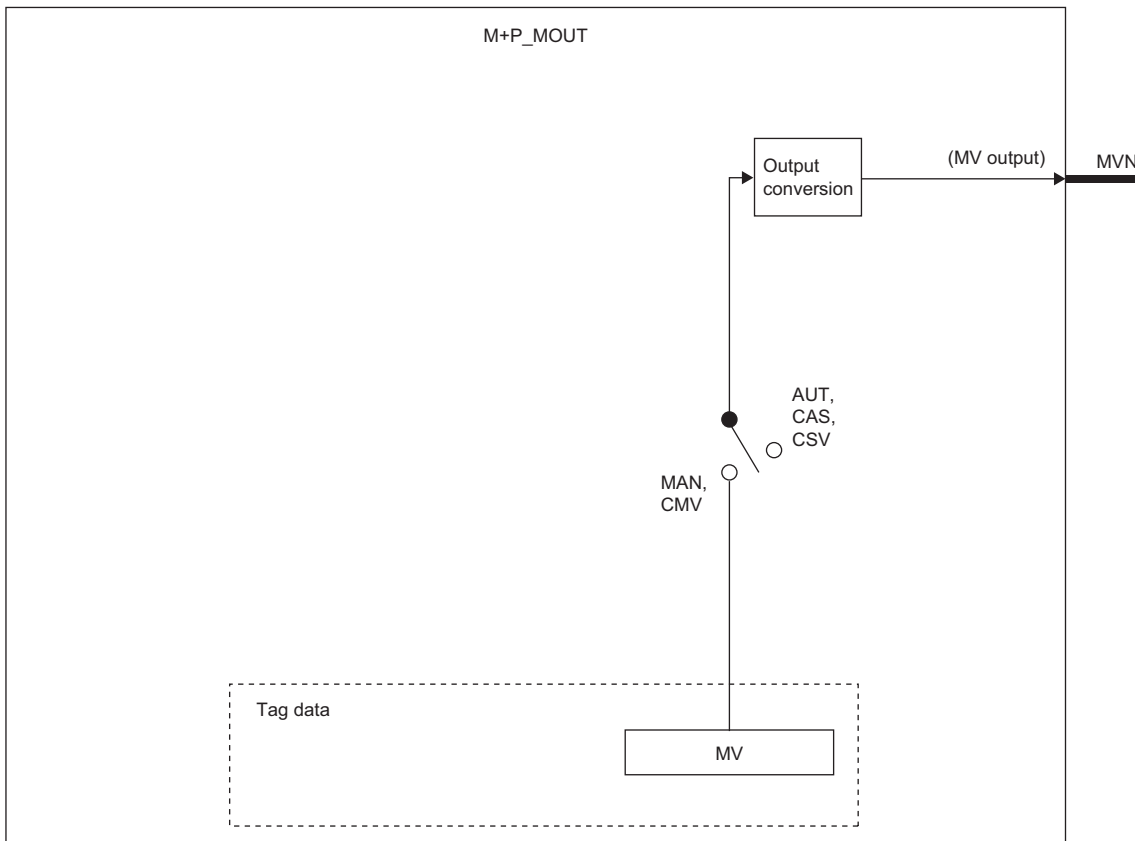
### Applicable tag type

MOUT, MWM

### Control mode

MAN	AUT	CAS	CMV	CSV
○	—	—	○	—

### Block diagram





## Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
MVN	Output to a module	NMIN to NMAX	Output variable	REAL

### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL

### Tag data

For details on tag data that is read or written with this tag access FB, refer to the following.

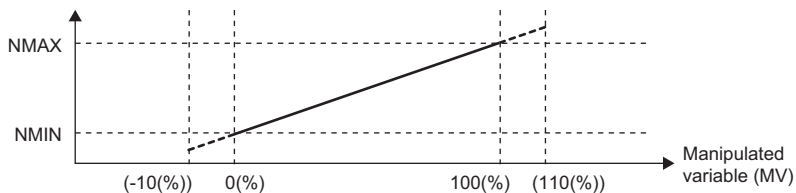
☞ Page 812 Tag Data List

## Processing details

### Output conversion

This function block performs output conversion processing.

Output of output conversion (MVN)



$$\text{Converted output (MVN)} = \left\{ (NMAX - NMIN) \times \frac{MV}{100} \right\} + NMIN$$

NMAX: Output conversion high limit value

NMIN: Output conversion low limit value

MV: Manipulated variable (%)

MVN: Output conversion output value

### Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation
	Output conversion
MAN, CMV	○
AUT, CAS, CSV	×

## Operation error

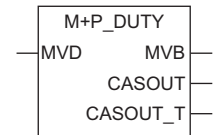
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Operation constant (NMAX, NMIN) or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# 14.6 Time Proportioning Output (M+P\_DUTY)

## M+P\_DUTY

This function block performs input addition, variation rate & high/low limiter, reset windup, output ON time conversion, and output conversion processing to an input value ( $\Delta MV$ ) and outputs the value as bits.

### FBD/LD



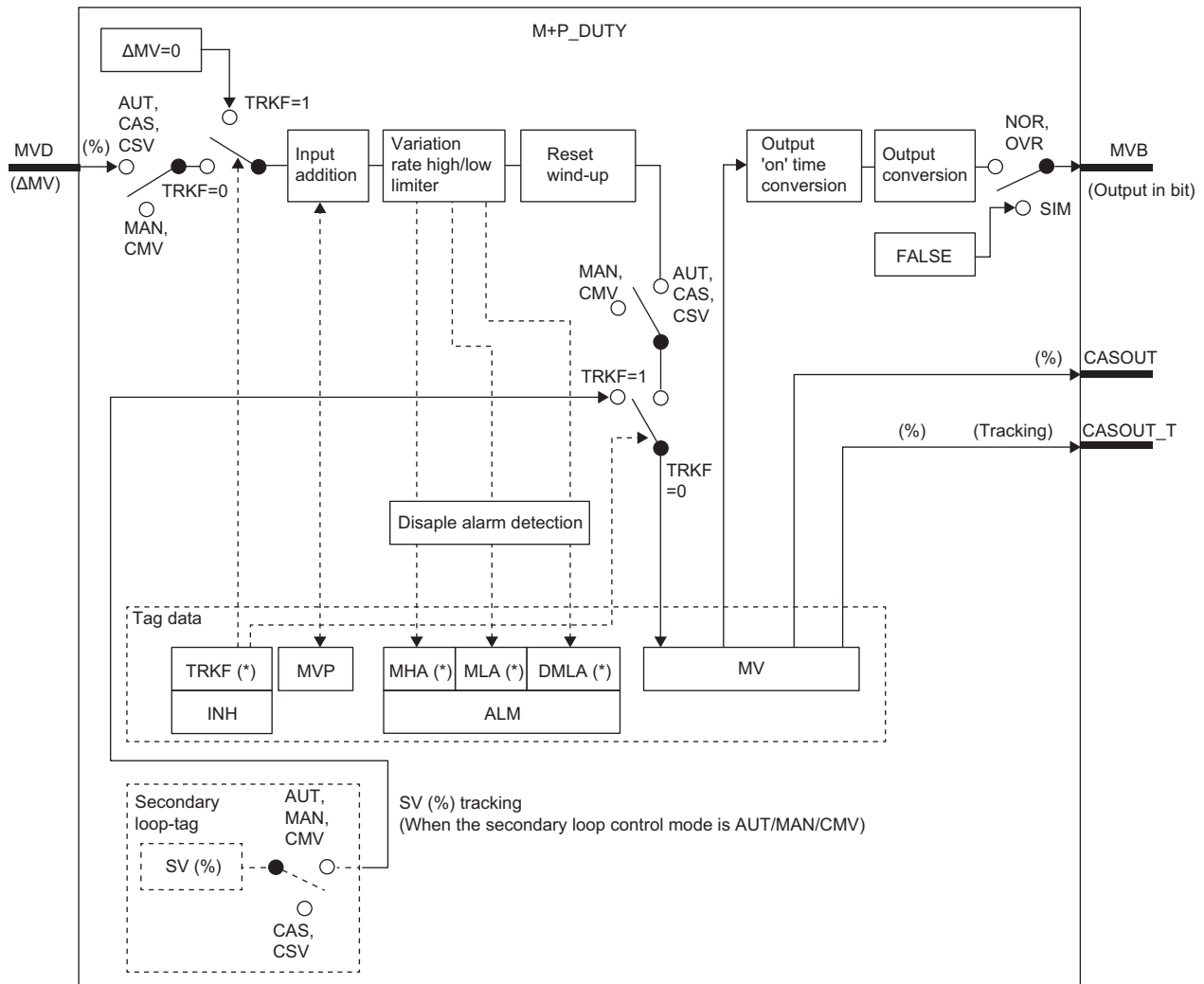
### Applicable tag type

BPI, IPD, PID, SPI, 2PID

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bits item.

## Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
MVD	$\Delta$ MV (%) input	-999999 to 999999	Input variable	REAL
MVB	Bit ON/OFF duty output to a module	TRUE, FALSE	Output variable	BOOL
CASOUT	Cascade output	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade output (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD


### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
SEA_OTYPE*1	Hold processing selection when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT

\*1 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

### Tag data

For details on tag data that is read or written with this tag access FB, refer to the following.

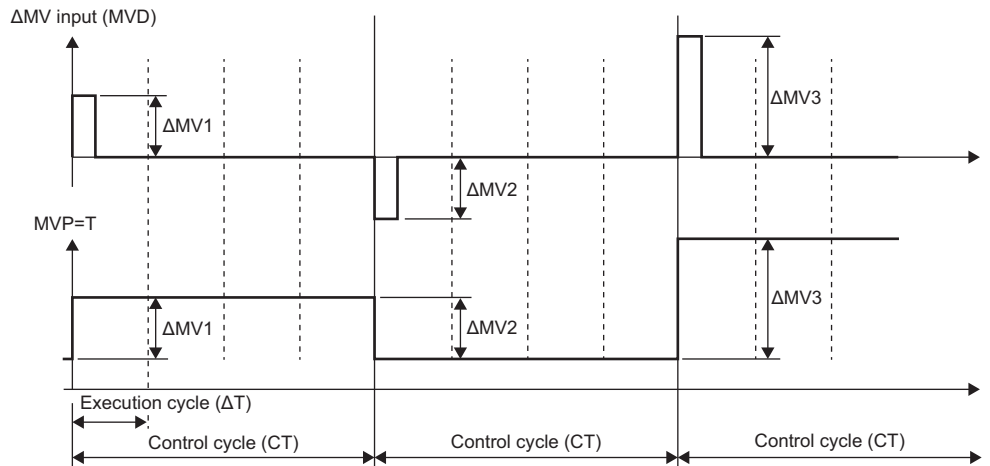
 Page 812 Tag Data List

## Processing details

### Input addition

A tentative manipulated value (T) is calculated from the input value ( $\Delta$ MV).

( $\Delta$ MV is output from an FB (such as M+P\_PID) before this FB every control cycle (CT).)



$$T = \Delta MV + MVP$$

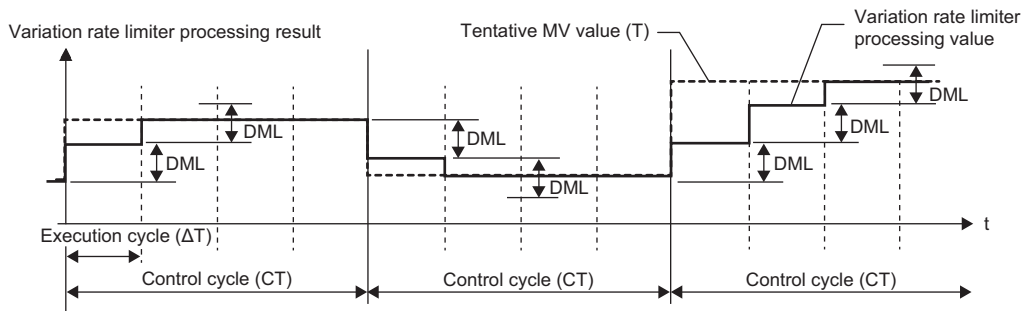
$$MVP = T$$

$\Delta$ MV: Input value to input (MVD), MVP: MV internal operation value, T: Tentative manipulated value

## Variation rate & high/low limiter

This function block checks the variation rate and high/low limits of the input value.

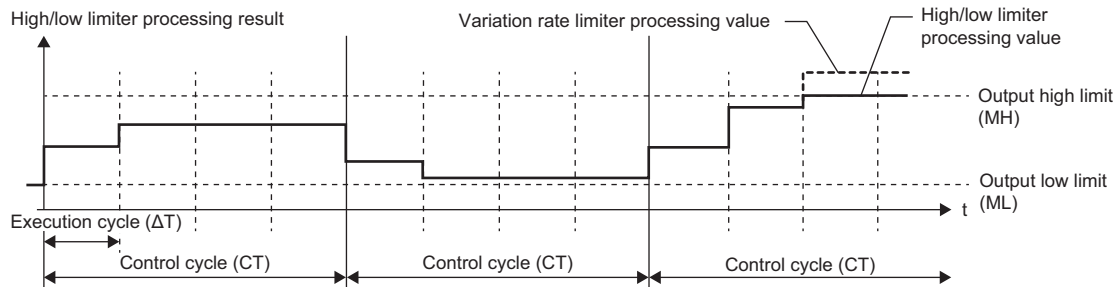
- Variation rate limiter



Condition	Variation rate limiter processing result	Alarm (ALM)
		Output variation rate limit (DMLA)
$ T - MV  \leq DML$	T	FALSE (Reset)
$T - MV > DML$	$MV + DML$	TRUE (Detected)
$T - MV < -DML$	$MV - DML$	TRUE (Detected)

T: Tentative manipulated value, MV: Manipulated value, DML: Output variation rate high limit value

- High/low limiter



Condition	High/low limiter processing result	Alarm (ALM)	
		Output low limit (MLA)	Output high limit (MHA)
Variation rate limiter processing result > MH	MH	FALSE (Reset)	TRUE (Detected)
Variation rate limiter processing result < ML	ML	TRUE (Detected)	FALSE (Reset)
$ML \leq$ Variation rate limiter processing result $\leq$ MH	Variation rate limiter processing value	FALSE (Reset)	FALSE (Reset)

MH: Output high limit value, ML: Output low limit value

## Reset windup

As countermeasures against reset windup, this function block sets a manipulated value to the high/low limit value when the manipulated value has exceeded the limit value, and quickly responds to a control target when deviation is inverted.

Condition	Countermeasure processing against reset windup
Variation rate limiter processing result > MH and $\frac{\Delta T}{T_i} \leq 1$	$MVP = \frac{\Delta T}{T_i} \times (MH - T) + T$
Variation rate limiter processing result < ML and $\frac{\Delta T}{T_i} \leq 1$	$MVP = \frac{\Delta T}{T_i} \times (ML - T) + T$

MH: Output high limit value  
 ML: Output low limit value  
 MVP: MV internal operation value  
 $\Delta T$ : Execution cycle  
 T: Tentative manipulated value  
 $T_i$ : Integral time

However, when  $T_i$  is 0, countermeasure processing against reset windup is not performed.

## Output ON time conversion/output conversion

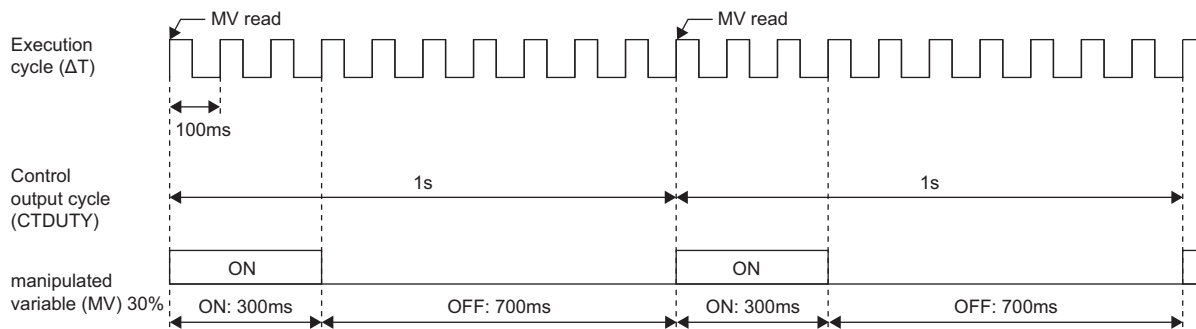
This function block outputs a duty manipulated value (MVB) to a manipulated value (MV).

Item	Description
Duty manipulated value (MVB) ON time	Duty manipulated value (MVB) ON time = ON time execution cycle count × ΔT The ON time execution cycle count is defined as the following value with the first digit after the decimal point rounded off. $\left( \frac{CTDUTY \times MV}{\Delta T \times 100} \right)$
Duty manipulated value (MVB) OFF time	Duty manipulated value (MVB) OFF time = OFF time execution cycle count × ΔT The OFF time execution cycle count is defined as (number of executions in a control output cycle) - (execution cycle count of ON time).

CTDUTY: Control output cycle, ΔT: Execution cycle, MV: Manipulated value (%)

### Ex.

Execution cycle: ΔT = 100ms, Control output cycle: CTDUTY = 1.0s, Manipulated value: MV = 30%



## Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the variation rate & high/low limiter processing.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no DMLA, MHA, or MLA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DMLI</li> <li>• MHI</li> <li>• MLI</li> </ul>
Disable alarm detection by control mode selection	In MAN or CMV mode, DMLA, MHA, and MLA of the alarm (ALM) are reset and DMLA, MHA, or MLA will not be detected.
Disable alarm detection by loop stop processing	Page 211 Loop stop processing

## Hold processing

This function block sets whether to hold M+P\_DUTY outputs or not according to the set value in the hold processing selection (SEA\_OTYPE) when a sensor error (SEA) has occurred in M+P\_IN, a tag access FB.

Condition	Processing result
SEA_OTYPE = 0	Comply with the setting common to all tags.*1 • "Yes" is selected for "Hold the output of M+P_OUT1, M+P_OUT2, M+P_DUTY"; Outputs are held. • "No" is selected for "Hold the output of M+P_OUT1, M+P_OUT2, M+P_DUTY"; The operation continues.
SEA_OTYPE = 1	Set the individual operation for each tag. Outputs are held.
SEA_OTYPE = 2	Set the individual operation for each tag. The operation continues.

\*1 The setting common to all tags is set in the process control extended setting in the engineering tool.

[Options] ⇒ [Convert] ⇒ [Process Control Extension Setting] ⇒ [I/O Control] ⇒ [Holding Processing]

## Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- Outputs (MVB) are held.
- The control mode is automatically switched to MANUAL.
- When DMLA, MHA, or MLA of the alarm (ALM) has occurred, the DMLA, MHA, or MLA is reset.
- No alarm is detected in the variation rate & high/low limiter processing.

## ■ Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation					
	Input processing	Variation rate & high/low limiter	Reset windup	Output ON time conversion	Output conversion	Alarm
MAN, CMV	×	×	×	○	○	×*1
AUT, CAS, CSV	○	○	○	○	○	○*2

\*1 An alarm (ALM) whose corresponding bit is TRUE (detected) is reset, and the alarm will not be detected.

\*2 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

## Operation error

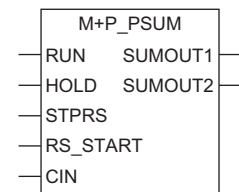
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (MVD) or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Control output cycle (CTDUTY) < 0
	The execution cycle (ΔT) setting is less than 0.
	(Control output cycle (CTDUTY) ÷ execution cycle (ΔT)) > 32767

# 14.7 Pulse Integrator (M+P\_PSUM)

## M+P\_PSUM

When the integration start signal (RUN) is TRUE, this function block performs input value increment operation, integrated value calculation, and output conversion processing to a count value (CIN) and outputs the result.

### FBD/LD



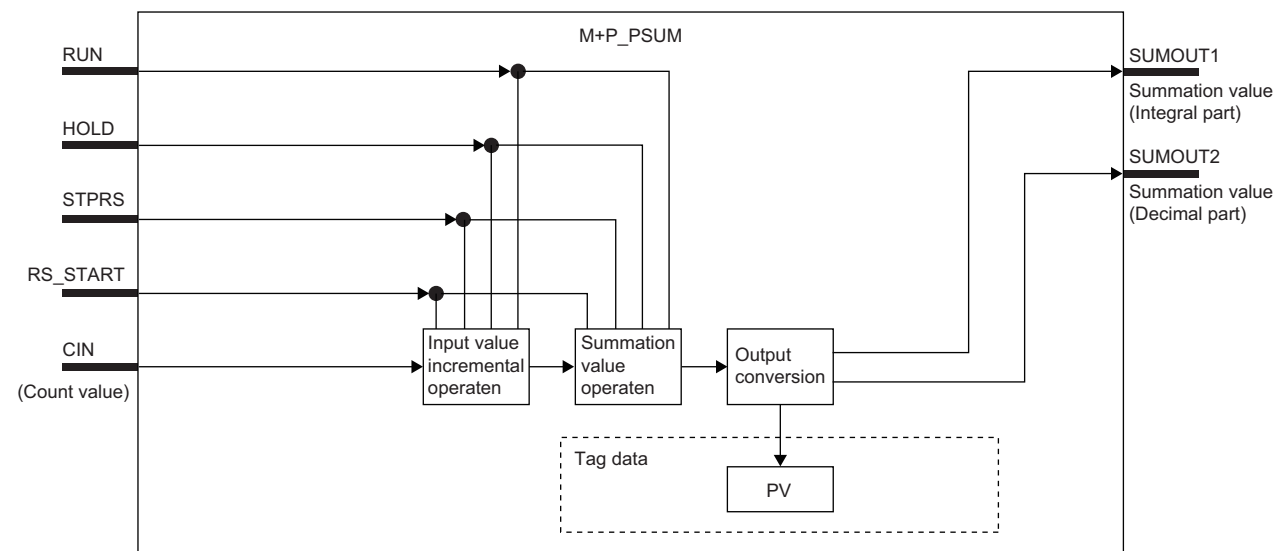
### Applicable tag type

PSUM, BC

### Control mode

MAN	AUT	CAS	CMV	CSV
—	—	—	—	—

### Block diagram



## Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
RUN	Integration start signal	TRUE: Execute FALSE: Stop	Input variable	BOOL
HOLD	Integration pause signal	TRUE: Execute FALSE: Stop	Input variable	BOOL
STPRS	Reset signal after integration pause	TRUE: Execute FALSE: Stop	Input variable	BOOL
RS_START	Start signal after integration reset	TRUE: Execute FALSE: Stop	Input variable	BOOL
CIN	Count value	Ring counter with the range of -2147483648 to 2147483647 (Pulse increment for each execution should be 32767 or less.)	Input variable	DINT
SUMOUT1	Integration value output (integral part)	0 to HILMT	Output variable	DINT
SUMOUT2	Integration value output (decimal part)	0 to 999	Output variable	DINT

### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
W_	Weight per pulse	1 to 999	1	User	INT
U_	Unit Conversion Constant	1, 10, 100, 1000	1	User	INT
HILMT	High Limit Value of Integration	1 to 2147483647	2147483647	User	DINT
SUMPTN	Integration Pattern	0: When an integrated value exceeds the integration high limit, the value is cleared to 0. 1: When an integrated value exceeds the integration high limit, the high limit value is held.	0	User	INT

### ■Tag data

For details on tag data that is read or written with this tag access FB, refer to the following.

☞ Page 812 Tag Data List

## Processing details

### ■Input condition

When the input variable RUN is TRUE, this function block performs integration processing to the input (CIN) and outputs the result.

When the input variable HOLD is TRUE, this function block holds integration processing to the input (CIN).

When the input variable STPRS is TRUE, this function block stops integration processing and resets the integrated value.

When the input variable RS\_START is TRUE, this function block resets integration processing and then restarts the integration processing.

### ■Input value increment operation

The following processing is performed for the input value (CIN).

Integration start signal (RUN)	Integration pause signal (HOLD)	Input value increment operation processing result (T1)
FALSE	FALSE	—
FALSE	TRUE	—
TRUE	FALSE	CIN - CIN <sub>n-1</sub>
TRUE	TRUE	—

CIN: Count value, CIN<sub>n-1</sub>: Last count value, T1: Input value increment operation processing result



### ■Integrated value calculation

This function block performs the following processing to the input value increment calculated by input value increment operation processing.

Integration start signal (RUN)	Integration pause signal (HOLD)	Integrated value calculation processing result (T2: Integrated value (integral part), T3: Integrated value (decimal part))
FALSE	FALSE	T2 = 0, T3 = 0
FALSE	TRUE	T2 = 0, T3 = 0
TRUE	FALSE	T4 = Quotient of $\{(T1 \times W\_)/U\_ \}$ (integral part) T5 = Remainder of $\{(T1 \times W\_)/U\_ \}$ (decimal part) T2 = Quotient of $PV + T4 + \{(SUM2 + T5)/U\_ \}$ (integral part) T3 = Remainder of $\{(SUM2 + T5)/U\_ \}$ (decimal part)
TRUE	TRUE	T2 = PV, T3 = SUM2

- T1: Input value increment operation processing result
- T2: Integrated value (integral part)
- T3: Integrated value (decimal part)
- T4: Integrated value increment (integral part)
- T5: Integrated value increment (decimal part)
- W\_: Weight per pulse
- U\_: Unit conversion constant
- PV: Integrated value (integral part)
- SUM2: Integrated value (decimal part)

### ■Output conversion

This function block performs the following processing to the integrated value calculated by integrated value calculation processing.

Integration pattern (SUMPTN)	Condition	Output variable (SUMOUT1, SUMOUT2)		Tag data (PV, SUM2)	
		Integrated value (integral part) (SUMOUT1)	Integrated value (decimal part) (SUMOUT2)	Integrated value (integral part) (PV)	Integrated value (decimal part) (SUM2)
0	T2 > HILMT	SUMOUT1 = T2 - HILMT - 1	SUMOUT2 = T3	PV = T2 - HILMT - 1	SUM2 = T3
	Others	SUMOUT1 = T2	SUMOUT2 = T3	PV = T2	SUM2 = T3
1	T2 > HILMT	SUMOUT1 = HILMT	SUMOUT2 = 0	PV = HILMT	SUM2 = 0
	Others	SUMOUT1 = T2	SUMOUT2 = T3	PV = T2	SUM2 = T3

- T2: Integrated value (integral part)
- T3: Integrated value (decimal part)
- PV: Integrated value (integral part)
- SUM2: Integrated value (decimal part)
- SUMOUT1: Integrated value output (integral part)
- SUMOUT2: Integrated value output (decimal part)

### ■Processing operation

The processing operation is not changed by the control mode.

### Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.

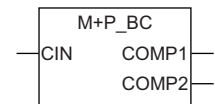
# 14.8 Batch Counter (M+P\_BC)

## M+P\_BC

This function block compares the input (CIN) with the estimated value 1 and estimated value 2, and outputs a completion signal when the input reaches an estimated value.

At this time, this function block performs the high limit check, variation rate check, and output conversion processing to the input (CIN).

### FBD/LD



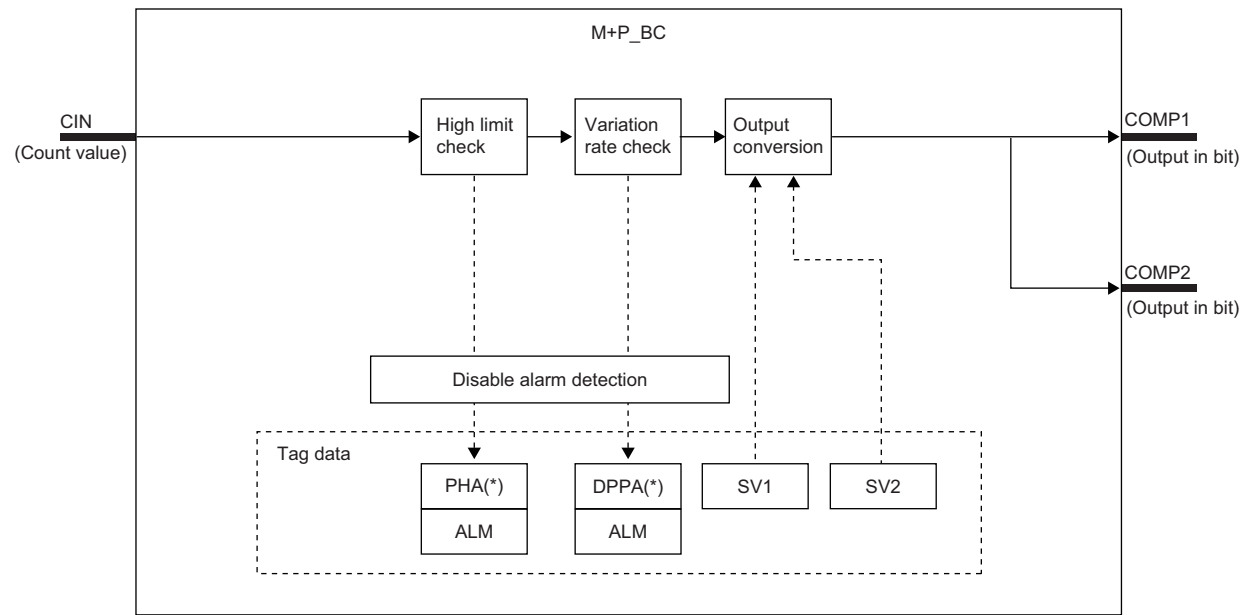
### Applicable tag type

BC

### Control mode

MAN	AUT	CAS	CMV	CSV
—	—	—	—	—

### Block diagram



\* Indicates bits item.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
CIN	Count value	0 to 99999999	Input variable	DINT
COMP1	Setting value 1 (SV1) completed output	TRUE: ON FALSE: OFF	Output variable	BOOL
COMP2	Setting value 2 (SV2) completed output	TRUE: ON FALSE: OFF	Output variable	BOOL

#### Tag data

For details on tag data that is read or written with this tag access FB, refer to the following.

📖 Page 863 BC

## Processing details

### High limit check

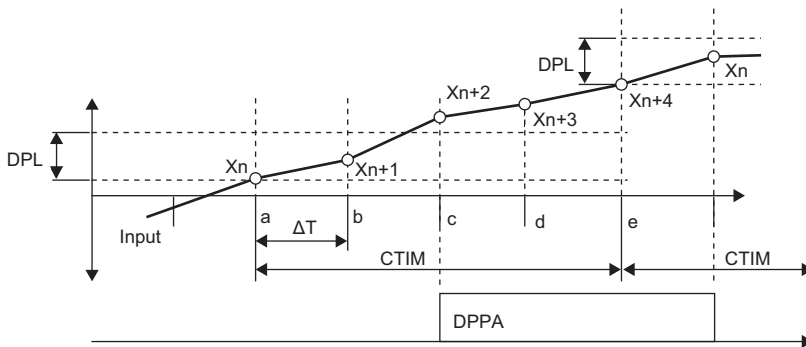
This function block performs the high limit check to the input (CIN).

Condition	Alarm (ALM)
	Input high limit (PHA)
$CIN > PH\_$	TRUE (Detected)
Others	FALSE (Reset)

CIN: Count value, PH\_: PV high limit alarm value

### Variation rate check

During the variation rate alarm check time (CTIM), this function block compares input changes with the variation rate alarm value (DPL) every execution cycle  $\Delta T$  and checks variation rate alarms.



In the example above, the variation rate b, c, d, e relative to  $X_n$  is checked and alarm output is exported.

- b:  $X_{n+1} - X_n$
- c:  $X_{n+2} - X_n$  (DPPA occurrence)
- d:  $X_{n+3} - X_n$  (DPPA occurrence)
- e:  $X_{n+4} - X_n$  (DPPA occurrence)

Condition	Alarm (ALM)
	Positive variation rate (DPPA)
$(X_n + m) - X_n \geq DPL$	TRUE (Detected)
Others	FALSE (Reset)

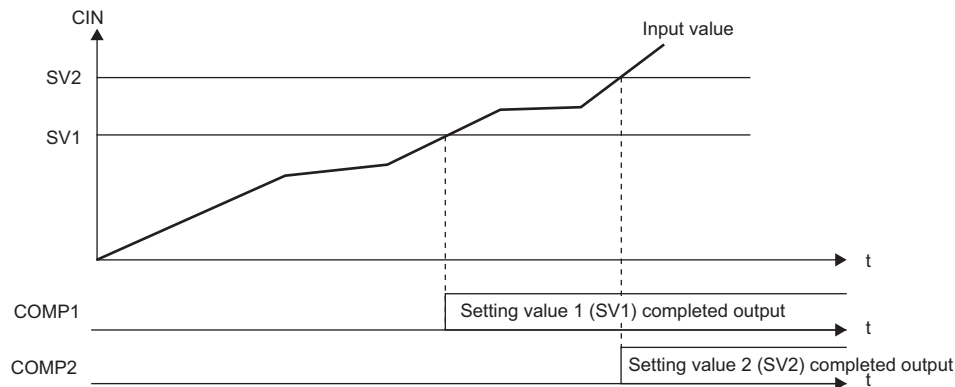
DPL: Variation rate alarm value (%)  
m: Variation rate monitoring counter

$$m = \frac{CTIM}{\Delta T}$$

$\Delta T$ : Execution cycle  
CTIM: Variation rate alarm check time

## Output conversion

This function block performs output conversion processing.



Condition	Setting value 1 (SV1) completed output (COMP1)	Setting value 2 (SV2) completed output (COMP2)
$CIN < 0$	FALSE	FALSE
$0 \leq CIN < SV1$	FALSE	—
$CIN \geq SV1$	TRUE	—
$0 \leq CIN < SV2$	—	FALSE
$CIN \geq SV2$	—	TRUE

CIN: Count value

SV1: Estimated value 1

SV2: Estimated value 2

COMP1: Setting value 1 (SV1) completed output value

COMP2: Setting value 2 (SV2) completed output value

## Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the high limit check and variation rate check.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	<p>If the following bits in the disable alarm detection (INH) are TRUE, no PHA or DPPA of the alarm (ALM) will be detected.</p> <ul style="list-style-type: none"> <li>• ERR1</li> <li>• PHI</li> <li>• DPPI</li> </ul>

## Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation			
	Input value increment operation	Integrated value calculation	Output conversion	Alarm
—	○	○	○	○*1

\*1 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

## Operation error

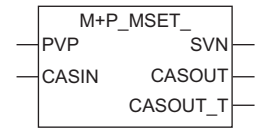
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (CIN) or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Variation rate alarm check time (CTIM) < 0
	The execution cycle (ΔT) setting is less than 0.
	(Variation rate alarm check time (CTIM) ÷ execution cycle (ΔT)) > 32767

# 14.9 Manual Setter (M+P\_MSET\_)

## M+P\_MSET\_

This function block performs SV variation rate & high/low limiter processing, sets the result as the set value (current) (SVC) of tag data, and outputs the value (SVC).

### FBD/LD



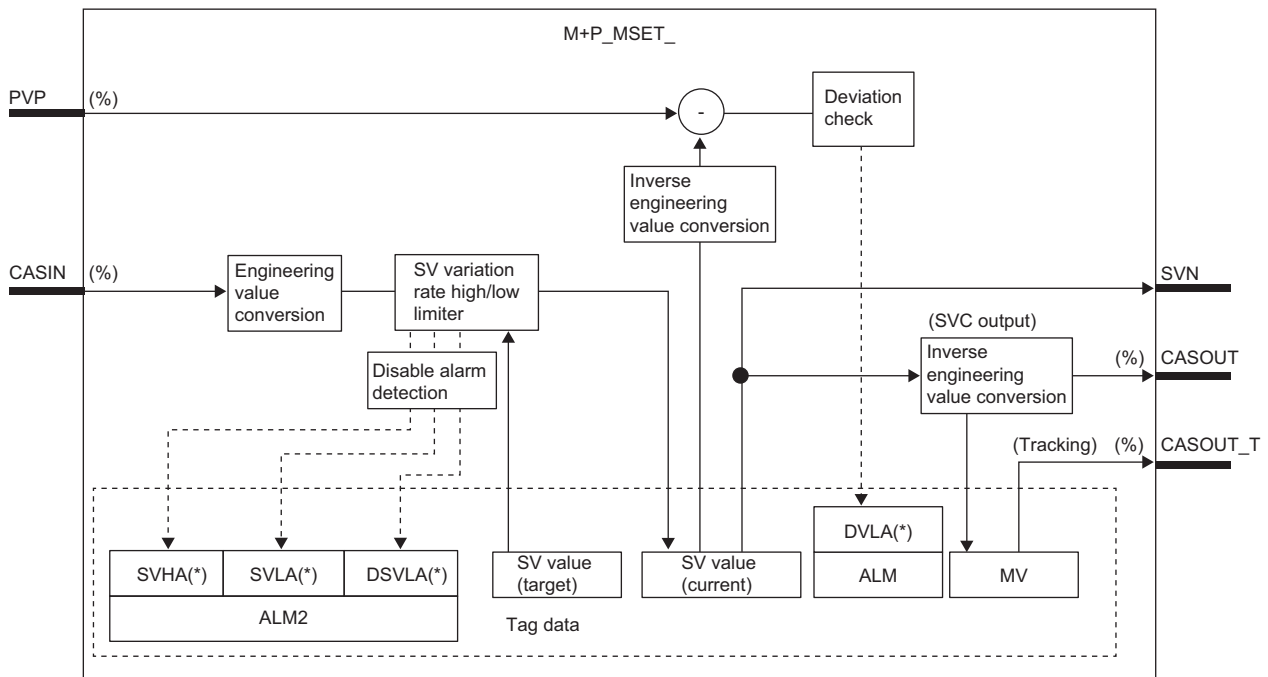
### Applicable tag type

SWM

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>

### Block diagram



\* Indicates bits item.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input	0 to 100[%]	Input variable	REAL
CASIN	Cascade SV input	0 to 100[%]	Input variable	REAL
SVN	Output to a module	-999999 to 999999	Output variable	REAL
CASOUT	Cascade SV output	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade SV output (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PN	Reverse action and direct action	0: Reverse action 1: Direct action	0	User	INT
SVPTN_B0	Setting Value (SV) Used <sup>*1</sup>	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SVLMT_EN	SV High/Low Limiter	TRUE: Execute FALSE: Not execute	FALSE	User	BOOL

\*1 When SVPTN\_B0 is TRUE, CASIN inputs cannot be used even though the mode is changed to the CAS mode.

## Tag data

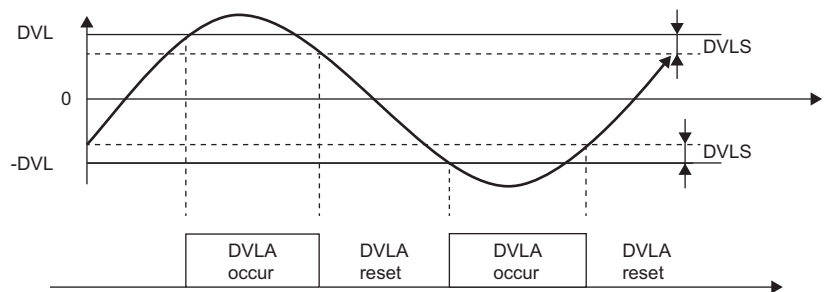
For details on tag data that is read or written with this tag access FB, refer to the following.

Page 857 SWM

## Processing details

### Deviation check

- This function block performs deviation check processing.



Condition	Alarm (ALM)
	Large deviation (DVLA)
$DVL <  DV $	TRUE (Detected)
$ DV  \leq (DVL - DVLS)$	FALSE (Reset)

DV: Deviation (%), DVLS: Large deviation alarm hysteresis (%), DVL: Deviation limit value (%)

- Deviation (DV) for direct/reverse action is calculated as follows.

Condition	Deviation (DV)
Direct action (PN = 1)	$DV (\%) = PVP (\%) - SVC (\%)$
Reverse action (PN = 0)	$DV (\%) = SVC (\%) - PVP (\%)$

DV: Deviation (%)

PVP (%): PV input value (%)

$$SVC(\%) = \frac{100}{RH-RL} \times (SVC-RL)$$

RH: Engineering value high limit

RL: Engineering value low limit

SVC: Setting value (current)

### Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = (RH - RL) / 100 \times \text{Setting value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

### ■SV variation rate & high/low limiter

This function block checks the variation rate and high/low limits to the set value every control cycle (CT).

- Variation rate limiter

The SV variation rate high limit value input in percentage is converted into an engineering value, and the processing will be performed.

DSVL → DSVLT (DSVL: SV variation rate high limit value, DSVLT: SV variation rate high limit value converted into an engineering value)

Condition	Variation rate limiter result	Alarm 2 (ALM2) SV variation rate limit alarm (DSVLA)
SV - SVC  ≤ DSVLT	SV	FALSE (Reset)
SV - SVC > DSVLT	SVC + DSVLT	TRUE (Detected)
SV - SVC < -DSVLT	SVC - DSVLT	TRUE (Detected)

SV: Setting value (target), SVC: Setting value (current)

If DSVLI of the disable alarm detection 2 or ERRI of the disable alarm detection is TRUE, DSVLA will be FALSE.

- High/low limiter

When SVLMT\_EN is TRUE

Condition	High/low limiter result	Alarm 2 (ALM2)	
		SV low limit (SVLA)	SV high limit (SVHA)
Variation rate limiter result > SH	SH	FALSE (Reset)	TRUE (Detected)
Variation rate limiter result < SL	SL	TRUE (Detected)	FALSE (Reset)
SL ≤ Variation rate limiter result ≤ SH	Variation rate limiter result	FALSE (Reset)	FALSE (Reset)

If SVLI of the disable alarm detection 2 or ERRI of the disable alarm detection is TRUE, SVLA will be FALSE.

If SVHI of the disable alarm detection 2 or ERRI of the disable alarm detection is TRUE, SVHA will be FALSE.

The high/low limiter result is stored in SVC (setting value (current)).

When SVLMT\_EN is FALSE

The variation rate limiter result is stored in SVC (set value (current)).

### ■Engineering value inverse conversion

This function block converts the setting value (SV) of an engineering value into a manipulated value in percentage (%).

$$MV(\%) = 100 / (RH - RL) \times (SV - RL)$$

RH: Engineering value high limit

RL: Engineering value low limit

SV: Setting value

MV: Manipulated variable

### ■Tracking processing

When tracking is requested by a subsequent FB through CASOUT\_T, this function block converts the value written to the MV into an engineering value and performs tracking to SV and SVC.

### ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) or the tag stop (TSTP) of the monitor output buffer (DOM) is TRUE, this function block performs the following processing.

- The control mode is automatically switched to MANUAL.
- When DVLA of the alarm (ALM) has occurred, the DVLA is reset. When DSVLA, SVLA, or SVHA of the alarm 2 (ALM2) has occurred, the DSVLA, SVLA, or SVHA is reset.
- No alarm is detected in the deviation check or SV variation rate & high/low limiter processing.

## ■ Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation*1			
	Deviation check	Engineering value conversion	Alarm	SV variation rate & high/low limiter
MAN, AUT	○	×	○*2	○*3
CAS, CSV	○	○	○*2	○

\*1 The processing of this tag access FB is performed every control cycle (CT).

\*2 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

\*3 In the MAN mode, SV variation rate limiter processing is not performed.

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Operation result of input data, operation constants, tag data, or inside of function blocks is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.



# 15 LOOP CONTROL OPERATION

The following FBs perform loop control processing including ratio control, various PID controls, two-position (on/off) control, three-position (on/off) control, program setter, and loop selector.

## Point

To set initial values of public variables of the tag access FB arranged on a user-defined tag FB in "FB Property" of the engineering tool, refer to the following.

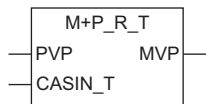
GX Works3 Operating Manual

## 15.1 Ratio Control (Enable Tracking for primary loop) (M+P\_R\_T)

### M+P\_R\_T

This FB controls two sets of control amount at a constant ratio and outputs the result ( $\Delta MV$ ). The primary loop can be tracked.

#### FBD/LD



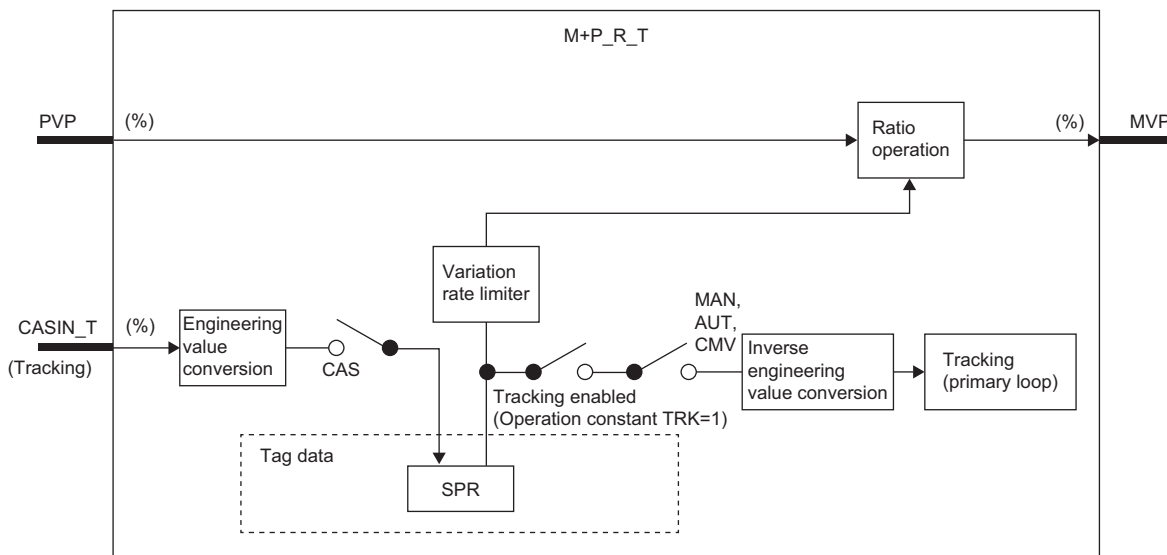
#### Applicable tag type

R

#### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

#### Block diagram



## Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN_T	Cascade SV input (unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVP	MV output (unit: %)	-999999 to 999999[%]	Output variable	REAL

### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
TRK	Tracking flag*1	0: Not executed 1: Executed	0	User	INT
SVPTN_B0	Setting value (SV) used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SVPTN_B1	Setting value (SV) pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

### Tag data

For details on tag data that is read or written with this tag access FB, refer to the following.

☞ Page 845 R

## Processing details

### Engineering value conversion

This function block performs engineering value conversion processing.

$$SPR = \frac{RMAX - RMIN}{100} \times CASIN + RMIN$$

SPR: Setting value (%)

RMIN: Ratio low limit (%)

RMAX: Ratio high limit (%)

CASIN: Setting value (%) from primary loop

### Variation rate limiter

This function block performs variation rate limiter processing.

Condition	Variation rate limiter processing result (Rn)
$(SPR - Rn) \geq DR$	$Rn = Rn-1 + DR$
$(SPR - Rn) \leq -DR$	$Rn = Rn-1 - DR$
$ SPR - Rn  < DR$	$Rn = SPR$

SPR: Setting value (%)

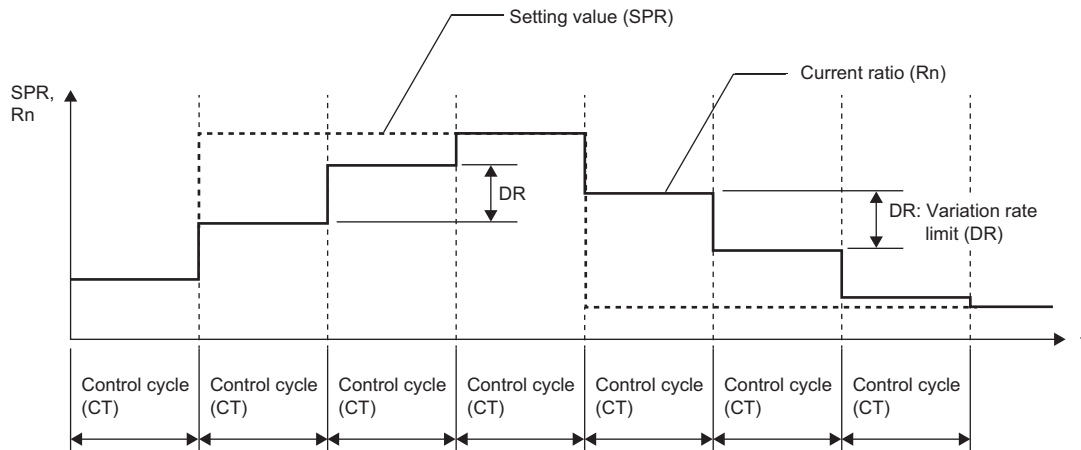
DR: Variation rate limit value (%)

Rn: Ratio current value (%)

Rn-1: Last value of ratio current value

## Ratio calculation

This function block performs ratio calculation processing.



$$\text{Ratio operation value (MVP)}(\%) = \frac{Rn - RMIN}{RMAX - RMIN} \times PVP + BIAS$$

Rn: Ratio current value (%)  
 RMIN: Ratio low limit (%)  
 RMAX: Ratio high limit (%)  
 PVP: PV input value (%)  
 BIAS: Bias

## Inverse engineering value conversion

This function block performs inverse engineering value conversion processing when the tracking flag (TRK) is 1.

$$SV(\%) = \frac{SPR - RMIN}{RMAX - RMIN} \times 100(\%)$$

SV: Tracking data (%) for primary loop  
 SPR: Setting value (%)  
 RMIN: Ratio low limit (%)  
 RMAX: Ratio high limit (%)

## Tracking processing

The following table shows whether tracking processing to the input variable CASIN\_T is performed or not.

Condition		Result
Tracking flag (TRK)	Setting value (SV) used (SVPTN_B0)	
1	FALSE	The input variable CASIN_T is tracked.
	TRUE	The input variable CASIN_T is not tracked.
0	FALSE or TRUE	

## Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- Outputs (MVP) are held.
- The control mode is automatically switched to MANUAL.

## Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation			
	Ratio calculation	Variation rate limiter	Engineering value conversion	Tracking
MAN, CMV	○	○	×	○*1
AUT, CAS, CSV	○	○	○	×

\*1 Tracking is performed when the tracking flag (TRK) is 1.

## Operation error

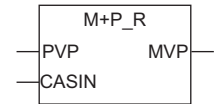
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP) or operation constant (TRK, SVPTN_B0, SVPTN_B1) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.

# 15.2 Ratio Control (Disable Tracking for primary loop) (M+P\_R)

## M+P\_R

This FB controls two sets of control amount at a constant ratio and outputs the result ( $\Delta MV$ ).

### FBD/LD



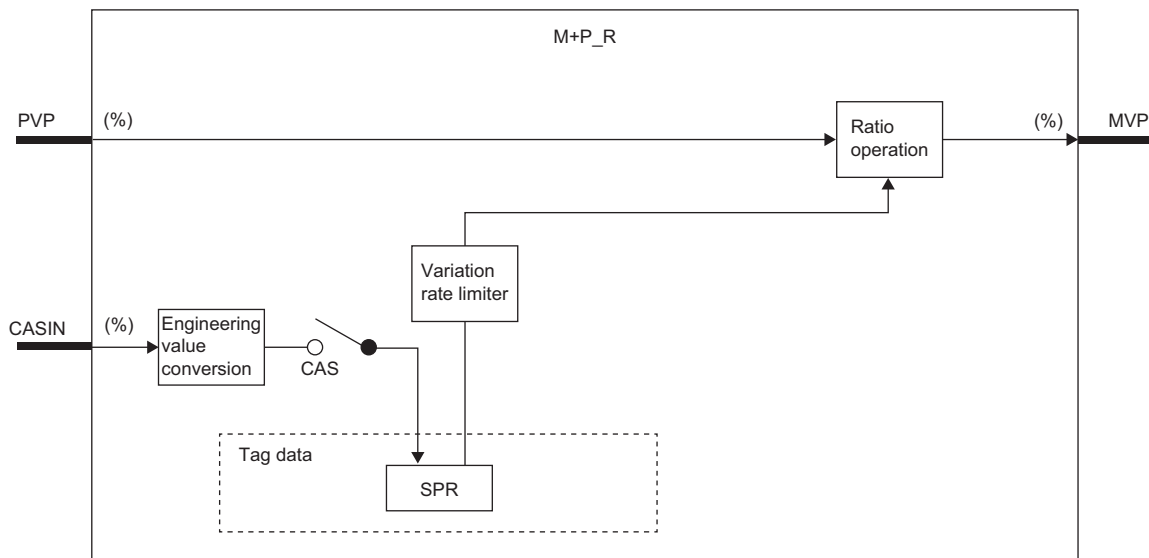
### Applicable tag type

R

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



15

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN	Cascade SV input (unit: %)	0 to 100[%]	Input variable	REAL
MVP	MV output (unit: %)	-999999 to 999999[%]	Output variable	REAL

#### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
SVPTN_B0	Setting value (SV) used	TRUE: Not used FALSE: Used	TRUE	User	BOOL

#### Tag data

For details on tag data that is read or written with this tag access FB, refer to the following.

☞ Page 845 R

## Processing details

### ■Engineering value conversion

This function block performs engineering value conversion processing.

$$SPR = \frac{RMAX-RMIN}{100} \times CASIN + RMIN$$

SPR: Setting value (%)

RMIN: Ratio low limit (%)

RMAX: Ratio high limit (%)

CASIN: Setting value (%) from primary loop

### ■Variation rate limiter

This function block performs variation rate limiter processing.

Condition	Variation rate limiter processing result (Rn)
$(SPR - Rn) \geq DR$	$Rn = Rn-1 + DR$
$(SPR - Rn) \leq -DR$	$Rn = Rn-1 - DR$
$ SPR - Rn  < DR$	$Rn = SPR$

SPR: Setting value (%)

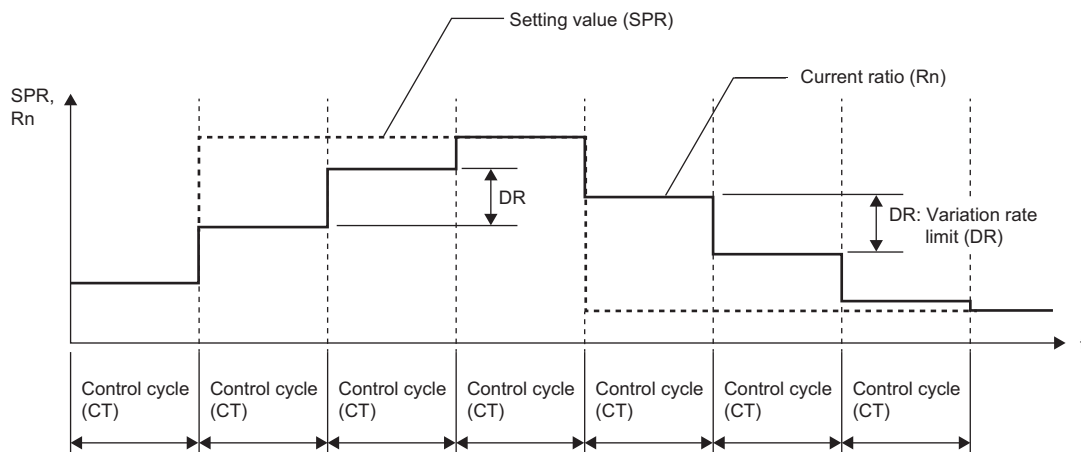
DR: Variation rate limit value (%)

Rn: Ratio current value (%)

Rn-1: Last value of ratio current value

### ■Ratio calculation

This function block performs ratio calculation processing.



$$\text{Ratio operation value (MVP)}(\%) = \frac{Rn-RMIN}{RMAX-RMIN} \times PVP + BIAS$$

Rn: Ratio current value (%)

RMIN: Ratio low limit (%)

RMAX: Ratio high limit (%)

PVP: PV input value (%)

BIAS: Bias

## ■ Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- Outputs (MVP) are held.
- The control mode is automatically switched to MANUAL.

## ■ Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation		
	Ratio calculation	Variation rate limiter	Engineering value conversion
MAN, CMV	○	○	×
AUT, CAS, CSV	○	○	○

## Operation error

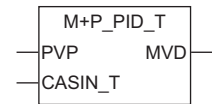
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP) or operation constant (SVPTN_B0) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.

# 15.3 Velocity Type PID Control (Enable Tracking for primary loop) (M+P\_PID\_T)

## M+P\_PID\_T

This FB performs PID operations using process variable differential, inexact differential, and velocity type, and outputs the result ( $\Delta MV$ ). The primary loop can be tracked.

### FBD/LD



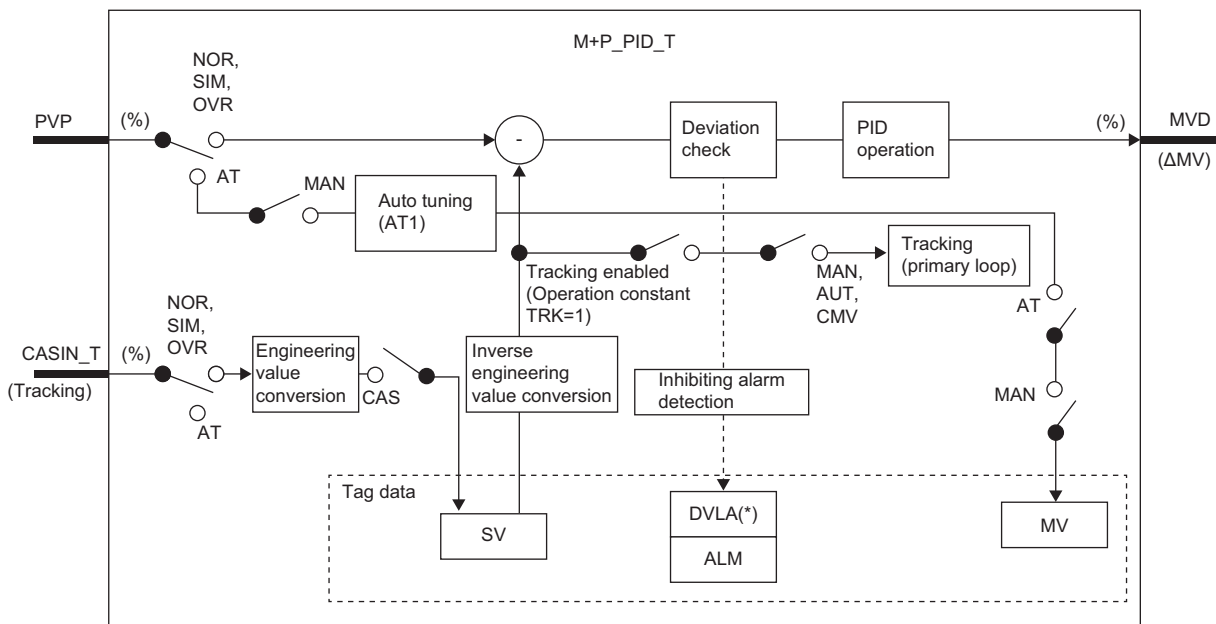
### Applicable tag type

PID

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bits item.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN_T	Cascade SV input (unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVD	$\Delta MV$ output (unit: %)	-999999 to 999999[%]	Output variable	REAL



## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MTD	Derivative Gain	0 to 9999	8.0	User	REAL
DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
SVPTN_B0	Setting value (SV) used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SVPTN_B1	Setting value (SV) pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

## Tag data

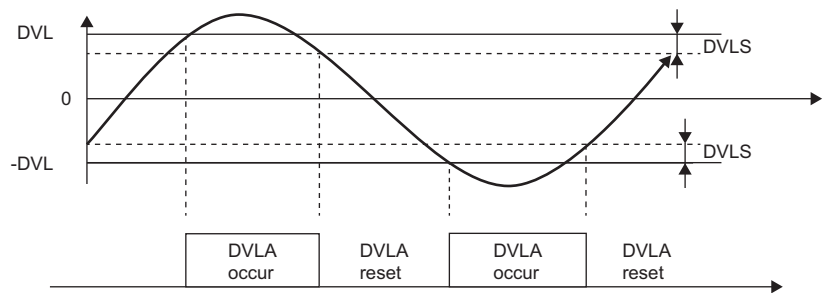
For details on tag data that is read or written with this tag access FB, refer to the following.

📖 Page 813 PID

## Processing details

### Deviation check

This function block performs deviation check processing.



Condition	Alarm (ALM)
	Large deviation (DVLA)
$DVL <  DV $	TRUE (Detected)
$ DV  \leq (DVL - DVLS)$	FALSE (Reset)

DV: Deviation (%), DVLS: Large deviation alarm hysteresis (%), DVL: Deviation limit value (%)

## ■PID operation

- Gain ( $K_p$ ) is calculated as follows.

$$K_p = K \times \text{PROPORTIONAL}$$

K: Output gain, PROPORTIONAL: Gain

- Output gain (K) is calculated as follows.

Condition	Output gain (K)
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is equal to 0</li> </ul>	K = 1
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is greater than 0</li> </ul>	When $ DV  \leq GW$ $K = GG$ When $ DV  > GW$ $K = 1 - \frac{(1-GG) \times GW}{ DV }$

DV: Deviation (%), GW: Gap width (%) = Rate of the gap width to the deviation, GG: Gap gain

- Deviation for PID operation ( $DV'$ ) is calculated as follows.

Condition	Deviation for PID operation ( $DV'$ )
$DV < -GW$	$DV' = -(GG \times GW) + (DV + GW)$
$ DV  \leq GW$	$DV' = GG \times DV$
$DV > GW$	$DV' = GG \times GW + (DV - GW)$

$DV'$ : Deviation for PID operation (%)

DV: Deviation (%)

GW: Gap width (%)

GG: Gap gain

- Deviation (DV) for direct/reverse action is calculated as follows.

Condition	Deviation (DV)
Direct action (PN = 1)	$DV (\%) = PVP (\%) - SV (\%)$
Reverse action (PN = 0)	$DV (\%) = SV (\%) - PVP (\%)$

DV: Deviation (%)

PVP (%): PV input value (%)

$$SV (\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit

RL: Engineering value low limit

SV: Setting value

- PID operations are conducted as follows.

Item	Direct action	Reverse action
Deviation (DVn)	$DVn = PVn - SVn$	$DVn = SVn - PVn$
Output variation ( $\Delta MV$ )	$\Delta MV = Kp \times \left\{ \underbrace{(DV_n - DV_{n-1})}_{\text{Proportional}} + \underbrace{\frac{CT}{Ti} \times DV_n}_{\text{Integral}} + \underbrace{B_n}_{\text{Derivative (imperfect derivative)}} \right\}$ <p>The following shows a proportional term, integral term, and derivative term of <math>\Delta MV</math>.</p> <p>■ Proportional term  <math>\Delta MV = Kp \times (DVn - DVn-1)</math></p> <p>■ Integral term  <math>\Delta MV = Kp \times \frac{CT}{Ti} \times DVn</math></p> <p>■ Derivative term  <math>\Delta MV = Kp \times Bn</math></p>	
Bn	$B_n = B_{n-1} + \frac{Md \times Td}{Md \times CT + Td} \times \{ (PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times B_{n-1}}{Td} \}$	$B_n = B_{n-1} + \frac{Md \times Td}{Md \times CT + Td} \times \{ -(PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times B_{n-1}}{Td} \}$

Kp: Gain  
Ti: Integral time  
Td: Derivative time  
Md: Derivative gain  
CT: Control cycle  
DVn: Deviation  
DVn-1: Last deviation value  
PVn: Process variable  
PVn-1: Last process variable  
PVn-2: Process variable before the last value  
SVn: Engineering value conversion processing result

The integral term and derivative term are as follows under the following conditions.

Item	Condition	Processing
Derivative term	When Td = 0	$Bn = 0$
	When the control mode is MAN	
	When the control mode is CMV	
Integral term	When Ti = 0	$\frac{CT}{Ti} \times DV_n = 0$
	When either of MH or ML error has occurred, MVP > MH and the following expression is satisfied $\frac{CT}{Ti} \times DV_n > 0$	
	When either of MH or ML error has occurred, MVP < ML and the following expression is satisfied $\frac{CT}{Ti} \times DV_n < 0$	

Ti: Integral time  
CT: Control cycle  
DVn: Deviation  
MH: Output high limit value  
ML: Output low limit value  
MVP: MV Internal operation value

- Set an integral multiple of the execution cycle ( $\Delta T$ ) as a control cycle (CT).
- Set 0.0 or a value equal to or larger than the control cycle (CT) as an integral constant.
- PID operations of this tag access FB are performed every control cycle (CT) ( $\Delta MV$  output).
- In other execution cycles ( $\Delta T$ ), the last value is held ( $\Delta MV = 0$ ).

### ■ Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH-RL}{100} \times \text{Setting value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■ Inverse engineering value conversion

This function block converts the setting value (SV) of an engineering value into a setting value (SV) in percentage (%).

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value


## ■ Tracking processing

The following table shows whether tracking processing to the input variable CASIN\_T is performed or not.


Condition		Result
Tracking Flag (TRK)	Setting Value (SV) Used (SVPTN_B0)	
1	FALSE	The input variable CASIN_T is tracked.
	TRUE	The input variable CASIN_T is not tracked.
0	FALSE or TRUE	

## ■ Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the deviation check.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no DVLA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DVLI</li> </ul>
Disable alarm detection by loop stop processing	 Page 234 Loop stop processing

## ■ Auto tuning (AT1)

This function block detects dynamic characteristics and automatically calculates a proportional gain (Kp), integral time (Ti), and derivative time (Td) for PID operations. ( Page 916 Auto tuning)

- Auto tuning is for setting initial values of the proportional gain (Kp), integral time (Ti), and derivative time (TD) for PID operations. The ZN method: Ziegler-Nichols' step response method is used for calculating the values.
- Auto tuning can be performed only in the MANUAL mode.

## ■ Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- ΔMV is cleared to 0.
- The control mode is automatically switched to MANUAL.
- When DVLA of the alarm (ALM) has occurred, the DVLA is reset.
- No alarm is detected in the deviation check.

## ■ Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation						
	Deviation check	PID operation	Engineering value conversion	Inverse engineering value conversion	Tracking	Alarm	Auto tuning
MAN, CMV, AUT	○	○	×	○	○*1	○*2	○*3
CAS, CSV	○	○	○	○	×	○*2	×

\*1 Tracking is performed when the tracking flag (TRK) is 1.

\*2 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

\*3 Auto tuning can be performed only in the MANUAL mode.

## Operation error

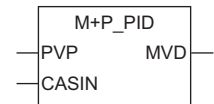
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP), operation constant (MTD, DVLS, PN, TRK, SVPTN_B0, SVPTN_B1), or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.
	The sampling interval time for AT1 (AT1ST) setting is less than 0.
	The time-out interval for AT1 (AT1TOUT1) setting is less than 0.
	The time-out interval after maximum slope for AT1 (AT1TOUT2) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the sampling interval time for AT1 (AT1ST) by the execution cycle ( $\Delta T$ ) exceeds 32767.
	The value divided the time-out interval for AT1 (AT1TOUT1) by the execution cycle ( $\Delta T$ ) exceeds 32767.
	The value divided the time-out interval after maximum slope for AT1 (AT1TOUT2) by the execution cycle ( $\Delta T$ ) exceeds 32767.

# 15.4 Velocity Type PID Control (Disable Tracking for primary loop) (M+P\_PID)

## M+P\_PID

This FB performs PID operations using process variable differential, inexact differential, and velocity type, and outputs the result ( $\Delta MV$ ).

### FBD/LD



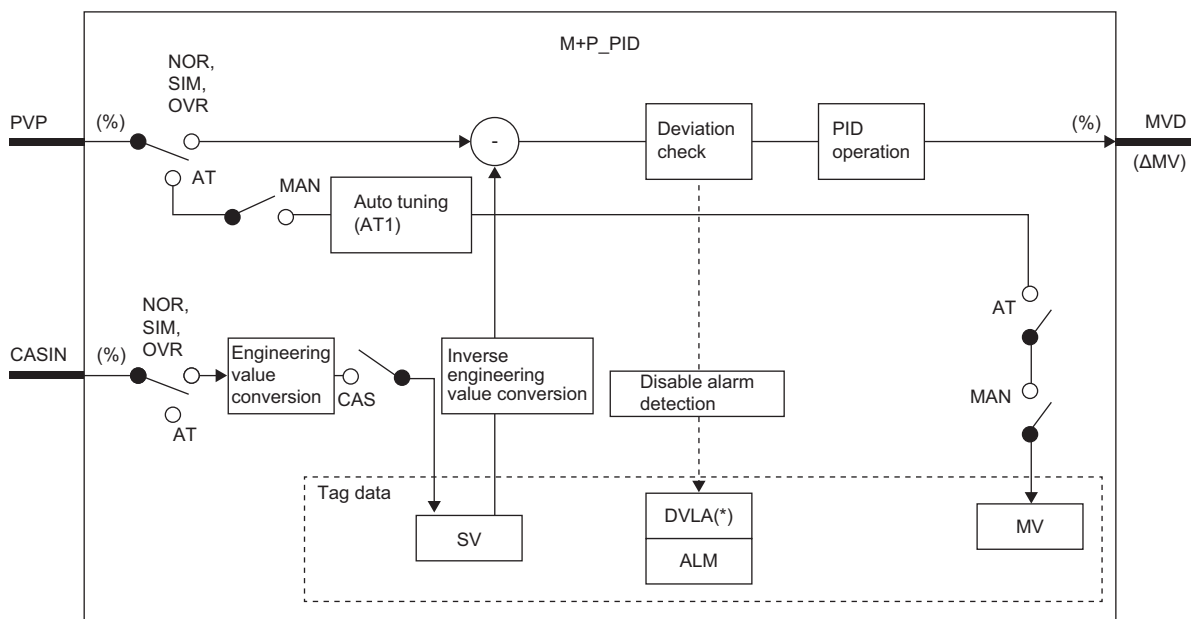
### Applicable tag type

PID

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bits item.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN	Cascade SV input (unit: %)	0 to 100[%]	Input variable	REAL
MVD	$\Delta MV$ output (unit: %)	-999999 to 999999[%]	Output variable	REAL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MTD	Derivative Gain	0 to 9999	8.0	User	REAL
DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
SVPTN_B0	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL

## Tag data

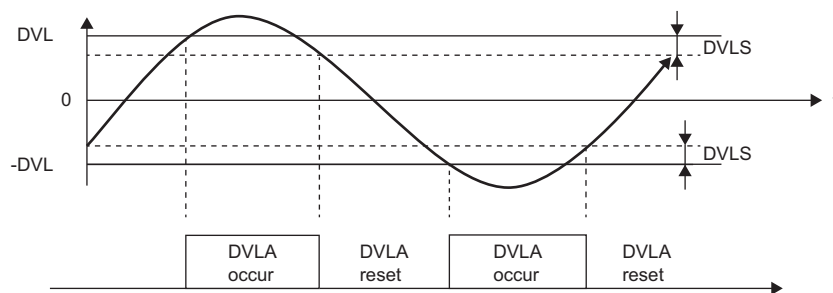
For details on tag data that is read or written with this tag access FB, refer to the following.

📖 Page 813 PID

## Processing details

### Deviation check

This function block performs deviation check processing.



Condition	Alarm (ALM)
	Large deviation (DVLA)
$DVL <  DV $	TRUE (Detected)
$ DV  \leq (DVL - DVLS)$	FALSE (Reset)

DV: Deviation (%), DVLS: Large deviation alarm hysteresis (%), DVL: Deviation limit value (%)

## ■PID operation

- Gain ( $K_p$ ) is calculated as follows.

$$K_p = K \times \text{PROPORTIONAL}$$

K: Output gain, PROPORTIONAL: Gain

- Output gain (K) is calculated as follows.

Condition	Output gain (K)
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is equal to 0</li> </ul>	K = 1
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is greater than 0</li> </ul>	When $ DV  \leq GW$ $K = GG$ When $ DV  > GW$ $K = 1 - \frac{(1-GG) \times GM}{ DV }$

DV: Deviation (%), GW: Gap width (%) = Rate of the gap width to the deviation, GG: Gap gain

- Deviation for PID operation ( $DV'$ ) is calculated as follows.

Condition	Deviation for PID operation ( $DV'$ )
$DV < -GW$	$DV' = -(GG \times GW) + (DV + GW)$
$ DV  \leq GW$	$DV' = GG \times DV$
$DV > GW$	$DV' = GG \times GW + (DV - GW)$

$DV'$ : Deviation for PID operation (%)

DV: Deviation (%)

GW: Gap width (%)

GG: Gap gain

- Deviation (DV) for direct/reverse action is calculated as follows.

Condition	Deviation (DV)
Direct action (PN = 1)	$DV (\%) = PVP (\%) - SV (\%)$
Reverse action (PN = 0)	$DV (\%) = SV (\%) - PVP (\%)$

DV: Deviation (%)

PVP (%): PV input value (%)

$$SV (\%) = \frac{100}{RH - RL} \times (SV - RL)$$

RH: Engineering value high limit

RL: Engineering value low limit

SV: Setting value



- PID operations are conducted as follows.

Item	Direct action	Reverse action
Deviation (DVn)	$DVn = PVn - SVn$	$DVn = SVn - PVn$
Output variation ( $\Delta MV$ )	$\Delta MV = K_p \times \left\{ (DV_n - DV_{n-1}) + \frac{CT}{T_i} \times DV_n + B_n \right\}$ <p style="text-align: center;"> <span style="margin-right: 40px;">Gain</span> <span style="margin-right: 40px;">Proportional</span> <span style="margin-right: 40px;">Integral</span> <span>Derivative (imperfect derivative)</span> </p> <p>The following shows a proportional term, integral term, and derivative term of <math>\Delta MV</math>.</p> <p>■ Proportional term  <math>\Delta MV = K_p \times (DV_n - DV_{n-1})</math></p> <p>■ Integral term  <math>\Delta MV = K_p \times \frac{CT}{T_i} \times DV_n</math></p> <p>■ Derivative term  <math>\Delta MV = K_p \times B_n</math></p>	
Bn	$B_n = B_{n-1} + \frac{Md \times T_d}{Md \times CT + T_d} \times \left\{ (PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times B_{n-1}}{T_d} \right\}$	$B_n = B_{n-1} + \frac{Md \times T_d}{Md \times CT + T_d} \times \left\{ -(PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times B_{n-1}}{T_d} \right\}$

- Kp: Gain
- Ti: Integral time
- Td: Derivative time
- Md: Derivative gain
- CT: Control cycle
- DVn: Deviation
- DVn-1: Last deviation value
- PVn: Process variable
- PVn-1: Last process variable
- PVn-2: Process variable before the last value
- SVn: Engineering value conversion processing result

The integral term and derivative term are as follows under the following conditions.

Item	Condition	Processing
Derivative term	When Td = 0	$B_n = 0$
	When the control mode is MAN	
	When the control mode is CMV	
Integral term	When Ti = 0	$\frac{CT}{T_i} \times DV_n = 0$
	When either of MH or ML error has occurred, MVP > MH and the following expression is satisfied $\frac{CT}{T_i} \times DV_n > 0$	
	When either of MH or ML error has occurred, MVP < ML and the following expression is satisfied $\frac{CT}{T_i} \times DV_n < 0$	

- Ti: Integral time
- CT: Control cycle
- DVn: Deviation
- MH: Output high limit value
- ML: Output low limit value
- MVP: MV Internal operation value

- Set an integral multiple of the execution cycle ( $\Delta T$ ) as a control cycle (CT).
- Set 0.0 or a value equal to or larger than the control cycle (CT) as an integral constant.
- PID operations of this tag access FB are performed every control cycle (CT) ( $\Delta MV$  output).
- In other execution cycles ( $\Delta T$ ), the last value is held ( $\Delta MV = 0$ ).

## ■Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH-RL}{100} \times \text{Setting value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Inverse engineering value conversion


This function block converts the setting value (SV) of an engineering value into a setting value (SV) in percentage (%).

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$


RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the deviation check.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no DVLA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DVLI</li> </ul>
Disable alarm detection by loop stop processing	 Page 240 Loop stop processing

## ■Auto tuning (AT1)

This function block detects dynamic characteristics and automatically calculates a proportional gain (Kp), integral time (Ti), and derivative time (Td) for PID operations. ( Page 916 Auto tuning)

- Auto tuning is for setting initial values of the proportional gain (Kp), integral time (Ti), and derivative time (TD) for PID operations. The ZN method: Ziegler-Nichols' step response method is used for calculating the values.
- Auto tuning can be performed only in the MANUAL mode.

## ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- ΔMV is cleared to 0.
- The control mode is automatically switched to MANUAL.
- When DVLA of the alarm (ALM) has occurred, the DVLA is reset.
- No alarm is detected in the deviation check.

## ■Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation					
	Deviation check	PID operation	Engineering value conversion	Inverse engineering value conversion	Alarm	Auto tuning
MAN, CMV, AUT	○	○	×	○	○ <sup>*1</sup>	○ <sup>*2</sup>
CAS, CSV	○	○	○	○	○ <sup>*1</sup>	×

\*1 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

\*2 Auto tuning can be performed only in the MANUAL mode.

## Operation error

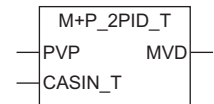
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP), operation constant (MTD, DVLS, PN, SVPTN_B0), or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.
	The sampling interval time for AT1 (AT1ST) setting is less than 0.
	The time-out interval for AT1 (AT1TOUT1) setting is less than 0.
	The time-out interval after maximum slope for AT1 (AT1TOUT2) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the sampling interval time for AT1 (AT1ST) by the execution cycle ( $\Delta T$ ) exceeds 32767.
	The value divided the time-out interval for AT1 (AT1TOUT1) by the execution cycle ( $\Delta T$ ) exceeds 32767.
	The value divided the time-out interval after maximum slope for AT1 (AT1TOUT2) by the execution cycle ( $\Delta T$ ) exceeds 32767.

# 15.5 2-degree-of-freedom PID Control (Enable Tracking for primary loop) (M+P\_2PID\_T)

## M+P\_2PID\_T

This FB optimizes the responsive performance (tracking performance) in a setting value change and control performance to a disturbance, and outputs the result ( $\Delta MV$ ). The primary loop can be tracked.

### FBD/LD



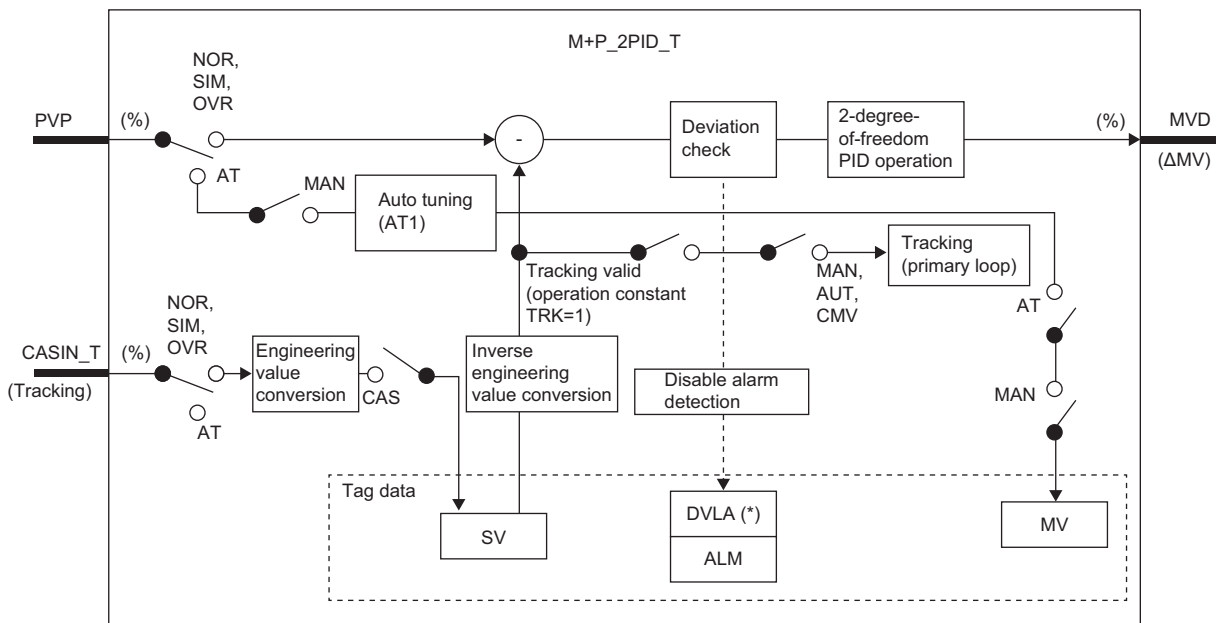
### Applicable tag type

2PID

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bits item.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN_T	Cascade SV input (unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVD	$\Delta MV$ output (unit: %)	-999999 to 999999[%]	Output variable	REAL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MTD	Derivative Gain	0 to 9999	8.0	User	REAL
DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
SVPTN_B0	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SVPTN_B1	Setting Value (SV) Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

## Tag data

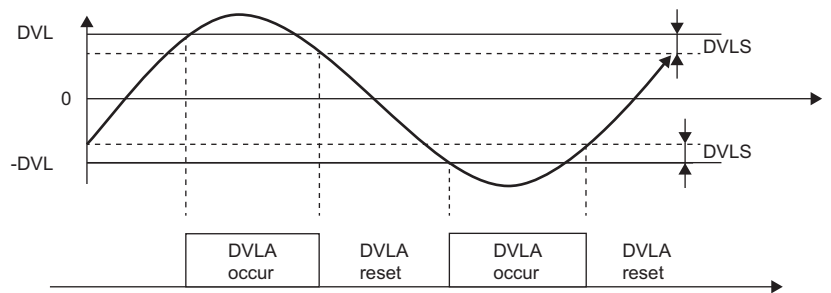
For details on tag data that is read or written with this tag access FB, refer to the following.

📖 Page 818 2PID

## Processing details

### Deviation check

This function block performs deviation check processing.



Condition	Alarm (ALM)
	Large deviation (DVLA)
$DVL <  DV $	TRUE (Detected)
$ DV  \leq (DVL - DVLS)$	FALSE (Reset)

DV: Deviation (%), DVLS: Large deviation alarm hysteresis (%), DVL: Deviation limit value (%)

### Two-degree-of-freedom PID operation

- Gain ( $K_p$ ) is calculated as follows.

$$K_p = K \times \text{PROPORTIONAL}$$

K: Output gain, PROPORTIONAL: Gain

- Output gain (K) is calculated as follows.

Condition	Output gain (K)
<ul style="list-style-type: none"> <li>• K value to the deviation (DV) of when the gap width (GW) is equal to 0</li> </ul>	K = 1
<ul style="list-style-type: none"> <li>• K value to the deviation (DV) of when the gap width (GW) is greater than 0</li> </ul>	When $ DV  \leq GW$ $K = GG$ When $ DV  > GW$ $K = 1 - \frac{(1-GG) \times GW}{ DV }$

DV: Deviation (%), GW: Gap width (%) = Rate of the gap width to the deviation, GG: Gap gain

- Deviation for PID operation (DV') is calculated as follows.

Condition	Deviation for PID operation (DV')
$DV < -GW$	$DV' = -(GG \times GW) + (DV + GW)$
$ DV  \leq GW$	$DV' = GG \times DV$
$DV > GW$	$DV' = GG \times GW + (DV - GW)$

DV': Deviation for PID operation (%)

DV: Deviation (%)

GW: Gap width (%)

GG: Gap gain

- Deviation (DV) for direct/reverse action is calculated as follows.

Condition	Deviation (DV)
Direct action (PN = 1)	$DV (\%) = PVP (\%) - SV (\%)$
Reverse action (PN = 0)	$DV (\%) = SV (\%) - PVP (\%)$

DV: Deviation (%)

PVP (%): PV input value (%)

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit

RL: Engineering value low limit

SV: Setting value

- PID operations are conducted as follows.

Item	Direct action	Reverse action
Deviation (DVn)	$DV_n = PV_n - SV_n$	$DV_n = SV_n - PV_n$
Output variation ( $\Delta MV$ )	$\Delta MV = K_p \times \left\{ (1 - \alpha) \times (DV_n - DV_{n-1}) + \frac{CT}{T_i} \times DV_n \right.$ <p style="text-align: center;"> <span style="margin-right: 100px;">Gain</span> <span style="margin-right: 100px;">Proportional</span> <span>Integral</span> </p> $+ (1 - \beta) \times B_n + \alpha \times C_n + \beta \times D_n \}$ <p style="text-align: center;"> <span style="margin-right: 100px;">Derivative</span> <span>Feed forward compensation</span> </p>	
Bn	$B_n = B_{n-1} + \frac{Md \times T_d}{Md \times CT + T_d} \times \left\{ (DV_n - 2DV_{n-1} + DV_{n-2}) - \frac{CT \times B_{n-1}}{T_d} \right\}$	
Dn	$D_n = D_{n-1} + \frac{Md \times T_d}{Md \times CT + T_d} \times \left\{ (PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times D_{n-1}}{T_d} \right\}$	$D_n = D_{n-1} + \frac{Md \times T_d}{Md \times CT + T_d} \times \left\{ -(PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times D_{n-1}}{T_d} \right\}$
Cn	$C_n = PV_n - PV_{n-1}$	$C_n = -(PV_n - PV_{n-1})$

- Kp: Gain
- Ti: Integral time
- Td: Derivative time
- Md: Derivative gain
- CT: Control cycle
- DVn: Deviation
- DVn-1: Last deviation value
- DVn-2: Deviation value before the last value
- PVn: Process variable
- PVn-1: Last process variable
- PVn-2: Process variable before the last value
- SVn: Engineering value conversion processing result
- $\alpha$ : Two-degree-of-freedom parameter (feedforward proportional)
- $\beta$ : Two-degree-of-freedom parameter (feedforward differential)

The integral term and derivative term are as follows under the following conditions.

Item	Condition	Processing
Derivative term	When Td = 0	Bn = 0
	When the control mode is MAN	
	When the control mode is CMV	
Integral term	When Ti = 0	$\frac{CT}{T_i} \times DV_n = 0$
	When either of MH or ML error has occurred, MVP > MH and the following expression is satisfied	
	When either of MH or ML error has occurred, MVP < ML and the following expression is satisfied	

- Ti: Integral time
- CT: Control cycle
- DVn: Deviation
- MH: Output high limit value
- ML: Output low limit value
- MVP: MV Internal operation value

Set an integral multiple of the execution cycle ( $\Delta T$ ) as a control cycle (CT).  
 Set 0.0 or a value equal to or larger than the control cycle (CT) as an integral constant.  
 PID operations of this tag access FB are performed every control cycle (CT) ( $\Delta MV$  output).  
 In other execution cycles ( $\Delta T$ ), the last value is held ( $\Delta MV = 0$ ).

## ■Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH-RL}{100} \times \text{Setting value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Inverse engineering value conversion

This function block converts the setting value (SV) of an engineering value into a setting value (SV) in percentage (%).

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value


## ■Tracking processing

The following table shows whether tracking processing to the input variable CASIN\_T is performed or not.


Condition		Result
Tracking Flag (TRK)	Setting Value (SV) Used (SVPTN_B0)	
1	FALSE	The input variable CASIN_T is tracked.
	TRUE	The input variable CASIN_T is not tracked.
0	FALSE or TRUE	

## ■Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the deviation check.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no DVLA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DVLI</li> </ul>
Disable alarm detection by loop stop processing	 Page 246 Loop stop processing

## ■Auto tuning (AT1)

This function block detects dynamic characteristics and automatically calculates a proportional gain (Kp), integral time (Ti), and derivative time (Td) for PID operations. ( Page 916 Auto tuning)

- Auto tuning is for setting initial values of the proportional gain (Kp), integral time (Ti), and derivative time (TD) for PID operations. The ZN method: Ziegler-Nichols' step response method is used for calculating the values.
- Auto tuning can be performed only in the MANUAL mode.

## ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- ΔMV is cleared to 0.
- The control mode is automatically switched to MANUAL.
- When DVLA of the alarm (ALM) has occurred, the DVLA is reset.
- No alarm is detected in the deviation check.



## ■ Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation						
	Deviation check	Two-degree-of-freedom PID operation	Engineering value conversion	Inverse engineering value conversion	Tracking	Alarm	Auto tuning
MAN, CMV, AUT	○	○	×	○	○*1	○*2	○*3
CAS, CSV	○	○	○	○	×	○*2	×

\*1 Tracking is performed when the tracking flag (TRK) is 1.

\*2 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

\*3 Auto tuning can be performed only in the MANUAL mode.

## Operation error

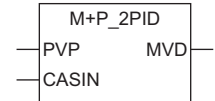
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP), operation constant (MTD, DVLS, PN, TRK, SVPTN_B0, SVPTN_B1), or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.
	The sampling interval time for AT1 (AT1ST) setting is less than 0.
	The time-out interval for AT1 (AT1TOUT1) setting is less than 0.
	The time-out interval after maximum slope for AT1 (AT1TOUT2) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the sampling interval time for AT1 (AT1ST) by the execution cycle ( $\Delta T$ ) exceeds 32767.
	The value divided the time-out interval for AT1 (AT1TOUT1) by the execution cycle ( $\Delta T$ ) exceeds 32767.
The value divided the time-out interval after maximum slope for AT1 (AT1TOUT2) by the execution cycle ( $\Delta T$ ) exceeds 32767.	

# 15.6 2-degree-of-freedom PID Control (Disable Tracking for primary loop) (M+P\_2PID)

## M+P\_2PID

This FB optimizes the responsive performance (tracking performance) in a setting value change and control performance to a disturbance, and outputs the result ( $\Delta MV$ ).

### FBD/LD



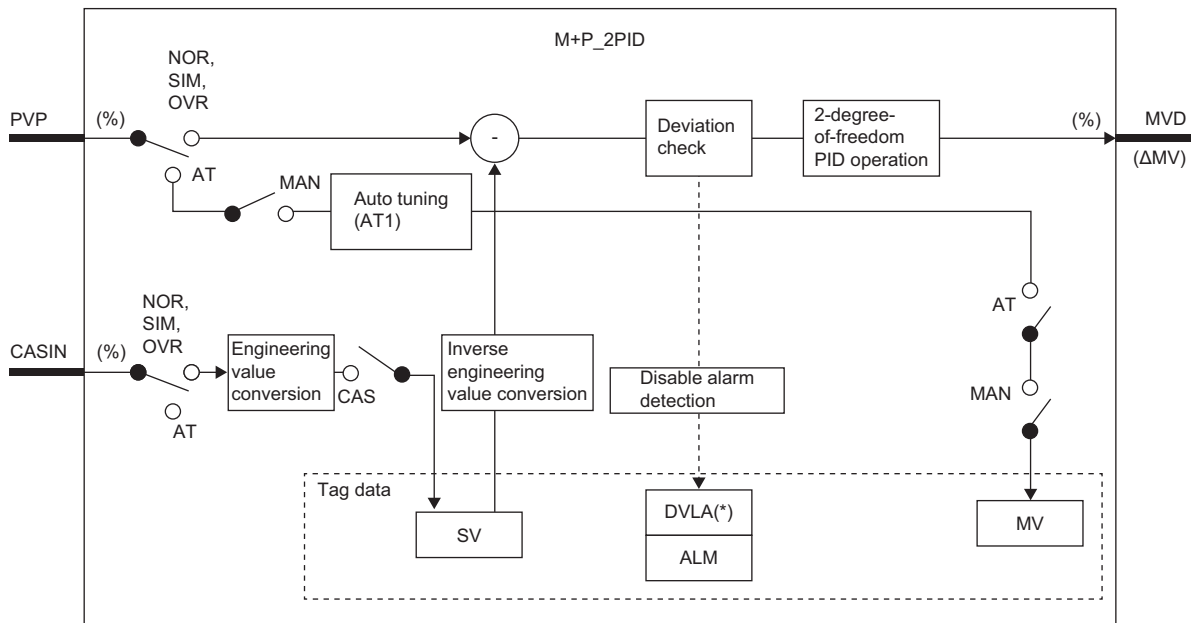
### Applicable tag type

2PID

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bits item.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN	Cascade SV input (unit: %)	0 to 100[%]	Input variable	REAL
MVD	$\Delta MV$ output (unit: %)	-999999 to 999999[%]	Output variable	REAL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MTD	Derivative Gain	0 to 9999	8.0	User	REAL
DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
SVPTN_B0	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL

## Tag data

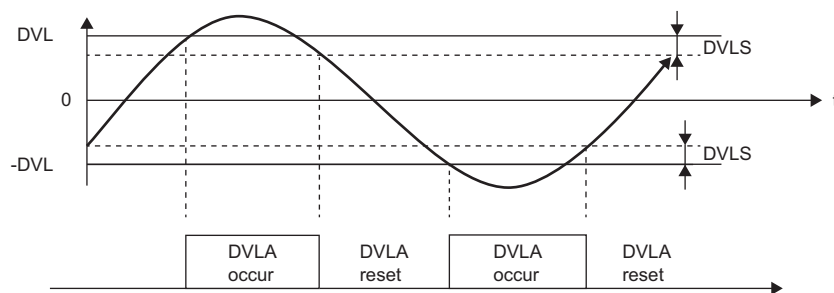
For details on tag data that is read or written with this tag access FB, refer to the following.

Page 818 2PID

## Processing details

### Deviation check

This function block performs deviation check processing.



Condition	Alarm (ALM)
	Large deviation (DVLA)
$DVL <  DV $	TRUE (Detected)
$ DV  \leq (DVL - DVLS)$	FALSE (Reset)

DV: Deviation (%), DVLS: Large deviation alarm hysteresis (%), DVL: Deviation limit value (%)

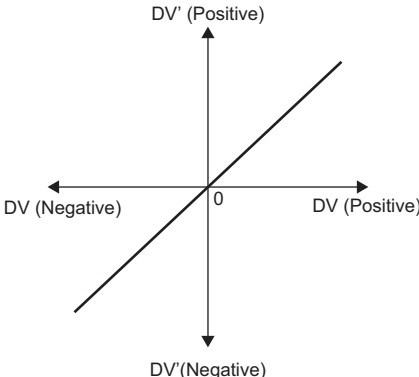
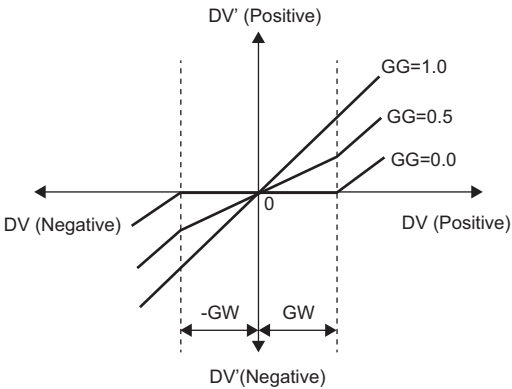
## Two-degree-of-freedom PID operation

- Gain ( $K_p$ ) is calculated as follows.

$$K_p = K \times \text{PROPORTIONAL}$$

K: Output gain, PROPORTIONAL: Gain

- Output gain (K) is calculated as follows.

Condition	Output gain (K)
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is equal to 0</li> </ul> 	K = 1
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is greater than 0</li> </ul> 	When $ DV  \leq GW$ $K = GG$ When $ DV  > GW$ $K = 1 - \frac{(1-GG) \times GW}{ DV }$

DV: Deviation (%), GW: Gap width (%) = Rate of the gap width to the deviation, GG: Gap gain

- Deviation for PID operation ( $DV'$ ) is calculated as follows.

Condition	Deviation for PID operation ( $DV'$ )
$DV < -GW$	$DV' = -(GG \times GW) + (DV + GW)$
$ DV  \leq GW$	$DV' = GG \times DV$
$DV > GW$	$DV' = GG \times GW + (DV - GW)$

$DV'$ : Deviation for PID operation (%)

DV: Deviation (%)

GW: Gap width (%)

GG: Gap gain

- Deviation (DV) for direct/reverse action is calculated as follows.

Condition	Deviation (DV)
Direct action (PN = 1)	$DV (\%) = PVP (\%) - SV (\%)$
Reverse action (PN = 0)	$DV (\%) = SV (\%) - PVP (\%)$

DV: Deviation (%)

PVP (%): PV input value (%)

$$SV (\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit

RL: Engineering value low limit

SV: Setting value

- PID operations are conducted as follows.

Item	Direct action	Reverse action
Deviation (DVn)	$DVn = PVn - SVn$	$DVn = SVn - PVn$
Output variation ( $\Delta MV$ )	$\Delta MV = \underbrace{Kp}_{\text{Gain}} \times \{ \underbrace{(1 - \alpha)}_{\text{Proportional}} \times (DV_n - DV_{n-1}) + \underbrace{\frac{CT}{Ti}}_{\text{Integral}} \times DV_n \}$ $+ \underbrace{(1 - \beta)}_{\text{Derivative}} \times B_n + \underbrace{\alpha \times C_n + \beta \times D_n}_{\text{Feed forward compensation}}$	
Bn	$B_n = B_{n-1} + \frac{Md \times Td}{Md \times CT + Td} \times \{ (DV_n - 2DV_{n-1} + DV_{n-2}) - \frac{CT \times B_{n-1}}{Td} \}$	
Dn	$D_n = D_{n-1} + \frac{Md \times Td}{Md \times CT + Td} \times \{ (PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times D_{n-1}}{Td} \}$	$D_n = D_{n-1} + \frac{Md \times Td}{Md \times CT + Td} \times \{ -(PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times D_{n-1}}{Td} \}$
Cn	$Cn = PVn - PVn-1$	$Cn = -(PVn - PVn-1)$

- Kp: Gain
- Ti: Integral time
- Td: Derivative time
- Md: Derivative gain
- CT: Control cycle
- DVn: Deviation
- DVn-1: Last deviation value
- DVn-2: Deviation value before the last value
- PVn: Process variable
- PVn-1: Last process variable
- PVn-2: Process variable before the last value
- SVn: Engineering value conversion processing result
- $\alpha$ : Two-degree-of-freedom parameter (feedforward proportional)
- $\beta$ : Two-degree-of-freedom parameter (feedforward differential)

The integral term and derivative term are as follows under the following conditions.

Item	Condition	Processing
Derivative term	When Td = 0	$Bn = 0$
	When the control mode is MAN	
	When the control mode is CMV	
Integral term	When Ti = 0	$\frac{CT}{Ti} \times DV_n = 0$
	When either of MH or ML error has occurred, MVP > MH and the following expression is satisfied	
	When either of MH or ML error has occurred, MVP < ML and the following expression is satisfied	

- Ti: Integral time
- CT: Control cycle
- DVn: Deviation
- MH: Output high limit value
- ML: Output low limit value
- MVP: MV Internal operation value

Set an integral multiple of the execution cycle ( $\Delta T$ ) as a control cycle (CT).  
 Set 0.0 or a value equal to or larger than the control cycle (CT) as an integral constant.  
 PID operations of this tag access FB are performed every control cycle (CT) ( $\Delta MV$  output).  
 In other execution cycles ( $\Delta T$ ), the last value is held ( $\Delta MV = 0$ ).

## ■Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH-RL}{100} \times \text{Setting value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Inverse engineering value conversion

This function block converts the setting value (SV) of an engineering value into a setting value (SV) in percentage (%).

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the deviation check.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no DVLA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DVLI</li> </ul>
Disable alarm detection by loop stop processing	☞ Page 252 Loop stop processing

## ■Auto tuning (AT1)

This function block detects dynamic characteristics and automatically calculates a proportional gain (Kp), integral time (Ti), and derivative time (Td) for PID operations. (☞ Page 916 Auto tuning)

- Auto tuning is for setting initial values of the proportional gain (Kp), integral time (Ti), and derivative time (TD) for PID operations. The ZN method: Ziegler-Nichols' step response method is used for calculating the values.
- Auto tuning can be performed only in the MANUAL mode.

## ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- ΔMV is cleared to 0.
- The control mode is automatically switched to MANUAL.
- When DVLA of the alarm (ALM) has occurred, the DVLA is reset.
- No alarm is detected in the deviation check.

## ■Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation					
	Deviation check	Two-degree-of-freedom PID operation	Engineering value conversion	Inverse engineering value conversion	Alarm	Auto tuning
MAN, CMV, AUT	○	○	×	○	○*1	○*2
CAS, CSV	○	○	○	○	○*1	×

\*1 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

\*2 Auto tuning can be performed only in the MANUAL mode.

## Operation error

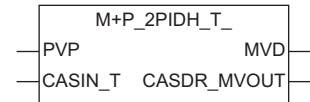
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP), operation constant (MTD, DVLS, PN, SVPTN_B0), or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.
	The sampling interval time for AT1 (AT1ST) setting is less than 0.
	The time-out interval for AT1 (AT1TOUT1) setting is less than 0.
	The time-out interval after maximum slope for AT1 (AT1TOUT2) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the sampling interval time for AT1 (AT1ST) by the execution cycle ( $\Delta T$ ) exceeds 32767.
	The value divided the time-out interval for AT1 (AT1TOUT1) by the execution cycle ( $\Delta T$ ) exceeds 32767.
	The value divided the time-out interval after maximum slope for AT1 (AT1TOUT2) by the execution cycle ( $\Delta T$ ) exceeds 32767.

# 15.7 2-degree-of-freedom Advanced PID Control (Enable Tracking for primary loop) (M+P\_2PIDH\_T\_)

## M+P\_2PIDH\_T\_

This FB optimizes the responsive performance (tracking performance) in a setting value change and control performance to a disturbance, and outputs the result ( $\Delta MV$ ). It also performs two-degree-of-freedom PID operation, PV tracking, integration stop, derivative stop, and SV variation rate & high/low limiter processing. The primary loop can be tracked.

### FBD/LD



### Applicable tag type

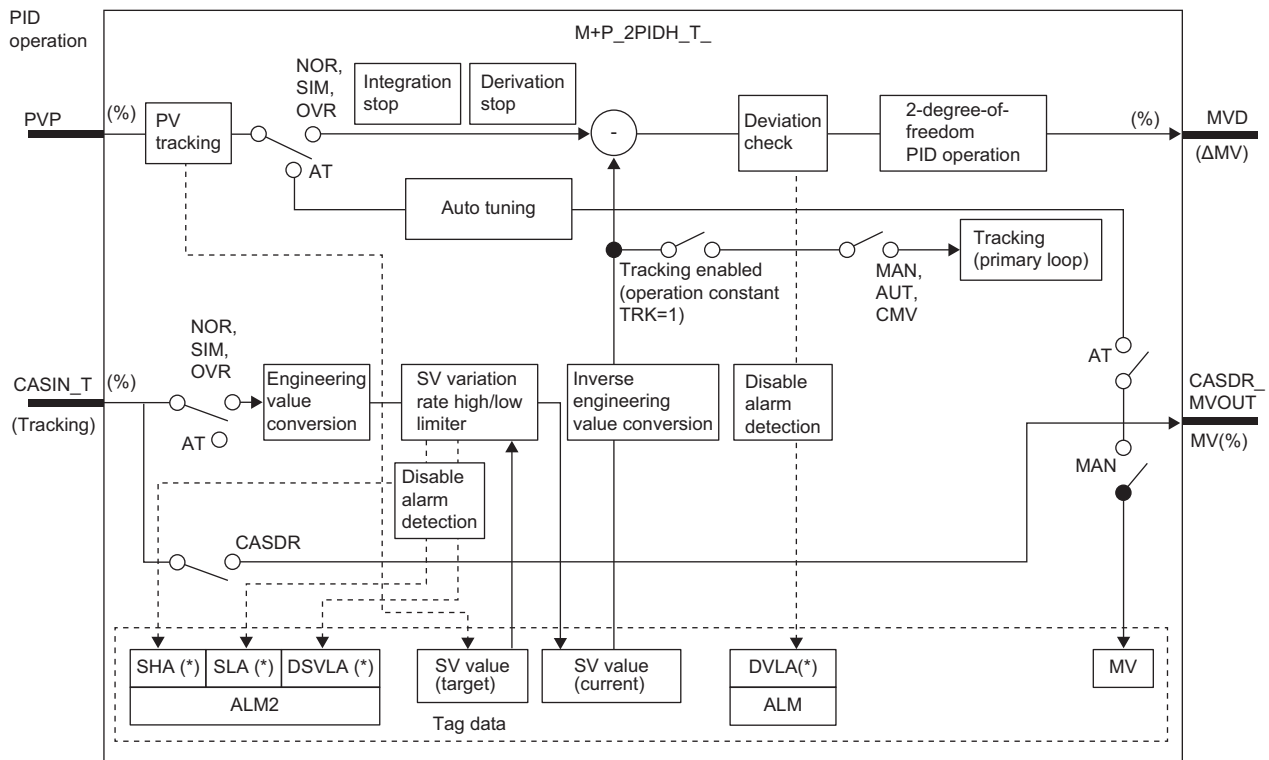
2PIDH

### Control mode

MAN	AUT	CAS* <sup>1</sup>	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

\*1 Transition to CASDR is possible.

### Block diagram



\* Indicates bits item.



## Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN_T	Cascade SV input (unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVD	$\Delta$ MV output (unit: %)	-999999 to 999999[%]	Output variable	REAL
CASDR_MVOUT	MV output for cascade direct (Unit: %)	0 to 100[%]	Output variable	REAL

### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MTD	Derivative Gain	0 to 9999	8.0	User	REAL
DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
SVPTN_B0	Setting value (SV) used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SVPTN_B1	Setting Value (SV) Pattern <sup>*1</sup>	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
PVTRK_EN	PV Tracking Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
ISTP	Integration Stop Signal	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
DSTP	Derivation Stop Signal	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
LMT_ISTP	Stop Integration, when MV variation rate limiter alarm occurred	TRUE: Stop FALSE: Not stop	FALSE	User	BOOL
SVLMT_EN	SV High/Low Limiter	TRUE: Executed FALSE: Not executed	FALSE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

### Tag data

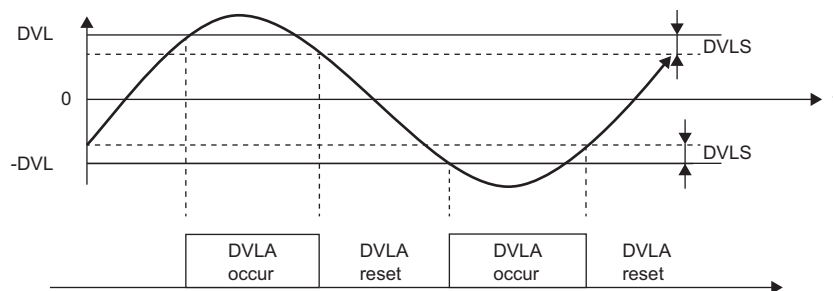
For details on tag data that is read or written with this tag access FB, refer to the following.

☞ Page 823 2PIDH

## Processing details

### Deviation check

This function block performs deviation check processing.



Condition	Alarm (ALM)
	Large deviation (DVLA)
$DVL <  DV $	TRUE (Detected)
$ DV  \leq (DVL - DVLS)$	FALSE (Reset)

DV: Deviation (%), DVLS: Large deviation alarm hysteresis (%), DVL: Deviation limit value (%)

## Two-degree-of-freedom PID operation

- Gain ( $K_p$ ) is calculated as follows.

$$K_p = K \times \text{PROPORTIONAL}$$

K: Output gain, PROPORTIONAL: Gain

- Output gain (K) is calculated as follows.

Condition	Output gain (K)	
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is equal to 0</li> </ul>	$K = 1$	
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is greater than 0</li> </ul>	<p>When <math> DV  \leq GW</math></p> <p>When <math> DV  &gt; GW</math></p>	<p><math>K = GG</math></p> <p><math>K = 1 - \frac{(1-GG) \times GW}{ DV }</math></p>

DV: Deviation (%), GW: Gap width (%) = Rate of the gap width to the deviation, GG: Gap gain

- Deviation for PID operation ( $DV'$ ) is calculated as follows.

Condition	Deviation for PID operation ( $DV'$ )
$DV < -GW$	$DV' = -(GG \times GW) + (DV + GW)$
$ DV  \leq GW$	$DV' = GG \times DV$
$DV > GW$	$DV' = GG \times GW + (DV - GW)$

$DV'$ : Deviation for PID operation (%)

DV: Deviation (%)

GW: Gap width (%)

GG: Gap gain

- Deviation (DV) for direct/reverse action is calculated as follows.

Condition	Deviation (DV)
Direct action (PN = 1)	$DV (\%) = PVP (\%) - SVC (\%)$
Reverse action (PN = 0)	$DV (\%) = SVC (\%) - PVP (\%)$

DV: Deviation (%)

PVP (%): PV input value (%)

$$SVC (\%) = \frac{100}{RH-RL} \times (SVC-RL)$$

RH: Engineering value high limit

RL: Engineering value low limit

SVC: Setting value (current)

- PID operations are conducted as follows.

Item	Direct action	Reverse action
Deviation (DVn)	$DV_n = PV_n - SV_n$	$DV_n = SV_n - PV_n$
Output variation ( $\Delta MV$ )	$\Delta MV = K_p \times \left\{ (1 - \alpha) \times (DV_n - DV_{n-1}) + \frac{CT}{T_i} \times DV_n \right.$ <p style="text-align: center;"> <span style="margin-right: 100px;">Gain</span> <span style="margin-right: 100px;">Proportional</span> <span>Integral</span> </p> $\left. + (1 - \beta) \times B_n + \alpha \times C_n + \beta \times D_n \right\}$ <p style="text-align: center;"> <span style="margin-right: 100px;">Derivative</span> <span>Feed forward compensation</span> </p>	
Bn	$B_n = B_{n-1} + \frac{Md \times T_d}{Md \times CT + T_d} \times \left\{ (DV_n - 2DV_{n-1} + DV_{n-2}) - \frac{CT \times B_{n-1}}{T_d} \right\}$	
Dn	$D_n = D_{n-1} + \frac{Md \times T_d}{Md \times CT + T_d} \times \left\{ (PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times D_{n-1}}{T_d} \right\}$	$D_n = D_{n-1} + \frac{Md \times T_d}{Md \times CT + T_d} \times \left\{ -(PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times D_{n-1}}{T_d} \right\}$
Cn	$C_n = PV_n - PV_{n-1}$	$C_n = -(PV_n - PV_{n-1})$

- Kp: Gain
- Ti: Integral time
- Td: Derivative time
- Md: Derivative gain
- CT: Control cycle
- DVn: Deviation
- DVn-1: Last deviation value
- DVn-2: Deviation value before the last value
- PVn: Process variable
- PVn-1: Last process variable
- PVn-2: Process variable before the last value
- SVn: Engineering value conversion processing result
- $\alpha$ : Two-degree-of-freedom parameter (feedforward proportional)
- $\beta$ : Two-degree-of-freedom parameter (feedforward differential)

The integral term and derivative term are as follows under the following conditions.

Item	Condition	Processing
Derivative term	When Td = 0	Bn = 0
	When the control mode is MAN	
	When the control mode is CMV	
Integral term	When Ti = 0	$\frac{CT}{T_i} \times DV_n = 0$
	When either of MH or ML error has occurred, MVP > MH and the following expression is satisfied	
	When either of MH or ML error has occurred, MVP < ML and the following expression is satisfied	

- Ti: Integral time
- CT: Control cycle
- DVn: Deviation
- MH: Output high limit value
- ML: Output low limit value
- MVP: MV Internal operation value

Set an integral multiple of the execution cycle ( $\Delta T$ ) as a control cycle (CT).  
 Set 0.0 or a value equal to or larger than the control cycle (CT) as an integral constant.  
 PID operations of this tag access FB are performed every control cycle (CT) ( $\Delta MV$  output).  
 In other execution cycles ( $\Delta T$ ), the last value is held ( $\Delta MV = 0$ ).

## ■Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH-RL}{100} \times \text{Setting value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value (target)

## ■Inverse engineering value conversion

This function block converts SVC of an engineering value into SVC in percentage (%).

$$SVC(\%) = \frac{100}{RH-RL} \times (SVC-RL)$$

RH: Engineering value high limit, RL: Engineering value low limit, SVC: Setting value (current)


## ■Tracking processing

The following table shows whether tracking processing to the input variable CASIN\_T is performed or not.

Condition		Result
Tracking Flag (TRK)	Setting Value (SV) Used (SVPTN_B0)	
1	FALSE	The input variable CASIN_T is tracked.
	TRUE	The input variable CASIN_T is not tracked.
0	FALSE or TRUE	

## ■Disable alarm detection

This function sets whether to detect alarms (ALM) or not in the deviation check and SV variation rate & high/low limiter processing.


Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) and the disable alarm detection 2 (INH2) are TRUE, DVLA of the alarm (ALM), or DSVLA, SVHA, or SVLA of the alarm 2 (ALM2) will not be detected. <ul style="list-style-type: none"> <li>• ERR1</li> <li>• DVLI</li> <li>• DSVLI</li> <li>• SVHI</li> <li>• SVLI</li> </ul>
Disable alarm detection by control mode selection	No alarm will be detected when the control mode is CASDR.
Disable alarm detection by loop stop processing	 Page 260 Loop stop processing

## ■Auto tuning (AT1, AT2)

This function block detects dynamic characteristics and automatically calculates a proportional gain (Kp), integral time (Ti), and derivative time (Td) for PID operations.


Select the step response method or the limit cycle method for auto tuning.

- AT1 (Step response method)

Auto tuning is for setting initial values of the proportional gain (Kp), integral time (Ti), and derivative time (Td) for PID operations. The ZN method: Ziegler-Nichols' step response method is used for calculating the values. ( Page 917 Step response method)

This method can be used in the MAN or CMV mode.

- AT2 (Limit cycle method)

In this method, a proportional gain (Kp), an integral time (Ti), and a derivative time (Td) for PID operations are calculated from the variation amplitude and variation cycle of a process variable obtained by repeatedly outputting the high and low limit values of a manipulated value. ( Page 920 Limit cycle method)

This method can be used in the MAN, AUT, CAS, CMV, or CSV mode.

## ■PV tracking function

To avoid sudden changes of the manipulated value at mode switching (from MAN to AUT), this function block matches the setting value (target) with the process variable when the control mode is MAN or CMV and keeps the value.

Condition	PV tracking processing
When PVTRK_EN is TRUE and the control mode is "MAN (CMV)"	Setting value (target) = Process variable Setting value (current) = Process variable
When PVTRK_EN is FALSE and the control mode is other than "MAN (CMV)"	Non-processing

## ■SV variation rate & high/low limiter

This function block checks the variation rate and high/low limits to the setting value every control cycle (CT).

- Variation rate limiter

When the control mode is AUT, CAS, or CSV

The SV variation rate high limit value input in percentage is converted into an engineering value, and the processing will be performed.

DSVL → DSVLT (DSVL: SV variation rate high limit value, DSVLT: SV variation rate high limit value converted into an engineering value)

Condition	Variation rate limiter result	Alarm 2 (ALM2) target variation rate limit (DSVLA)
$ SV - SVC  \leq DSVLT$	SV	FALSE (Reset)
$SV - SVC > DSVLT$	$SVC + DSVLT$	TRUE (Detected)
$SV - SVC < -DSVLT$	$SVC - DSVLT$	TRUE (Detected)

SV: Setting value (target), SVC: Setting value (current)

If DSVLI of the disable alarm detection 2 or ERRI of the disable alarm detection is TRUE, DSVLA will be FALSE.

When the control mode is MAN, CMV, or CASDR

Condition	Variation rate limiter result	Alarm 2 (ALM2) target variation rate limit (DSVLA)
None	SV	FALSE (Reset)

- High/low limiter

When the control mode is MAN, AUT, CAS, CMV, or CSV and SVLMT\_EN is TRUE

Condition	High/low limiter result	Alarm 2 (ALM2)	
		SV low limit (SVLA)	SV high limit (SVHA)
Variation rate limiter result > SH	SH	FALSE (Reset)	TRUE (Detected)
Variation rate limiter result < SL	SL	TRUE (Detected)	FALSE (Reset)
$SL \leq \text{Variation rate limiter result} \leq SH$	Variation rate limiter result	FALSE (Reset)	FALSE (Reset)

If SVLI of the disable alarm detection 2 or ERRI of the disable alarm detection is TRUE, SVLA will be FALSE.

If SVHI of the disable alarm detection 2 or ERRI of the disable alarm detection is TRUE, SVHA will be FALSE.

The high/low limiter result is stored in SVC (setting value (current)).

When the control mode is CASDR, or when SVLMT\_EN is FALSE

The variation rate limiter result is stored in SVC (set value (current)).

## ■Integration stop

This function block stops an integral element operation.

Condition	Processing
ISTP = TRUE	The integral element operation is stopped.
ISTP = FALSE	Non-processing

## Derivative stop

This function block stops a derivative element operation.

Condition	Processing
DSTP = TRUE	The derivative element operation is stopped.
DSTP = FALSE	Non-processing

## Integration stop at MV variation rate limiter occurrence

This function block stops an integral element operation when an MV variation rate limiter alarm occurs.

Condition	Processing
When LMT_ISTP is TRUE and a DMLA alarm has occurred	When the variation rate of a manipulated value has exceeded a positive/negative limit value and the sign (positive/negative) of an integral operation value is the same as that of the limit value, the integral element operation is stopped.
LMT_ISTP = FALSE	Non-processing

## Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) or the tag stop (TSTP) of the monitor output buffer (DOM) is TRUE, this function block performs the following processing.

- $\Delta MV$  is cleared to 0.
- The control mode is automatically switched to MANUAL.
- When DVLA of the alarm (ALM) has occurred, the DVLA is reset. When DSVLA, SVLA, or SVHA of the alarm 2 (ALM2) has occurred, the DSVLA, SVLA, or SVHA is reset.
- No alarm is detected in the deviation check or SV variation rate & high/low limiter processing.

## Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation					
	Deviation check	Two-degree-of-freedom PID operation	Engineering value conversion	Inverse engineering value conversion	Alarm	Auto tuning (AT1)
MAN, CMV	○	○	×	○	○ <sup>*1</sup>	○
AUT	○	○	×	○	○ <sup>*1</sup>	×
CAS, CSV	○	○	○	○	○ <sup>*1</sup>	×
CASDR	×	○	○	○	×	×

Control mode	Processing operation				
	Auto tuning (AT2)	PV tracking	SV variation rate & high/low limiter	Integration stop	Derivative stop
MAN, CMV	○	○	○ <sup>*2*3</sup>	○	○
AUT	○	×	○ <sup>*3</sup>	○	○
CAS, CSV	○	×	○ <sup>*3</sup>	○	○
CASDR	×	×	×	○	○

\*1 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

\*2 In the MAN mode, SV variation rate limiter processing is not performed.

\*3 This processing is not performed when a sensor error (SEA) has occurred and output hold has been selected. SVC (setting value (current)) is also not updated.

## Operation error

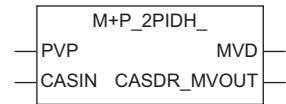
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data, operation constant, tag data, or operation result within a function block is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The sampling interval time for AT1 (AT1ST) setting is less than 0.
	The time-out interval for AT1 (AT1TOUT1) setting is less than 0.
	The time-out interval after maximum slope for AT1 (AT1TOUT2) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the sampling interval time for AT1 (AT1ST) by the execution cycle ( $\Delta T$ ) exceeds 32767.
	The value divided the time-out interval for AT1 (AT1TOUT1) by the execution cycle ( $\Delta T$ ) exceeds 32767.
	The value divided the time-out interval after maximum slope for AT1 (AT1TOUT2) by the execution cycle ( $\Delta T$ ) exceeds 32767.
	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.

# 15.8 2-degree-of-freedom Advanced PID Control (Disable Tracking for primary loop) (M+P\_2PIDH\_)

## M+P\_2PIDH\_

This FB optimizes the responsive performance (tracking performance) in a setting value change and control performance to a disturbance, and outputs the result ( $\Delta MV$ ). It also performs two-degree-of-freedom PID operation, PV tracking, integration stop, derivative stop, and SV variation rate & high/low limiter processing.

### FBD/LD



### Applicable tag type

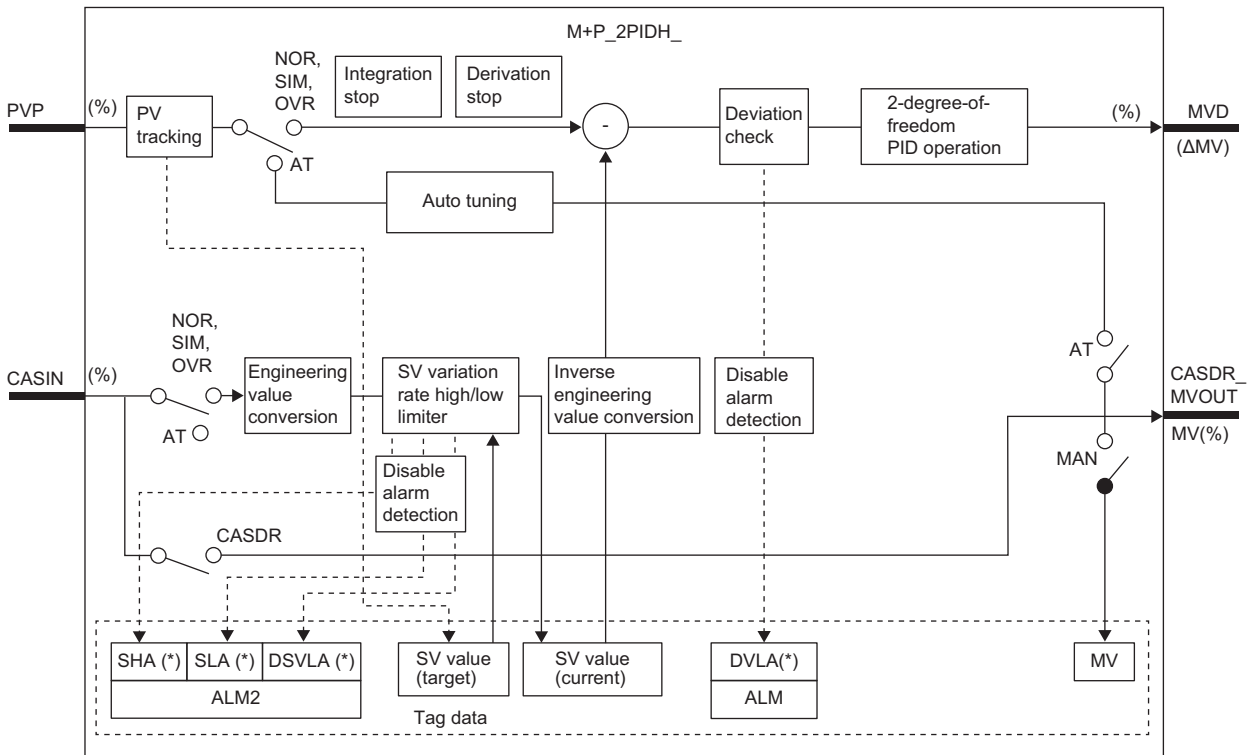
2PIDH

### Control mode

MAN	AUT	CAS*1	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

\*1 Transition to CASDR is possible.

### Block diagram



\* Indicates bits item.



## Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN	Cascade SV input (unit: %)	0 to 100[%]	Input variable	REAL
MVD	$\Delta$ MV output (unit: %)	-999999 to 999999[%]	Output variable	REAL
CASDR_MVOOUT	MV output for cascade direct (Unit: %)	0 to 100[%]	Output variable	REAL

### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MTD	Derivative Gain	0 to 9999	8.0	User	REAL
DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
SVPTN_B0	Setting value (SV) used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
PVTRK_EN	PV Tracking Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
ISTP	Integration Stop Signal	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
DSTP	Derivation Stop Signal	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
LMT_ISTP	Stop Integration, when MV variation rate limiter alarm occurred	TRUE: Stop FALSE: Not stop	FALSE	User	BOOL
SVLMT_EN	SV High/Low Limiter	TRUE: Executed FALSE: Not executed	FALSE	User	BOOL

15

### Tag data

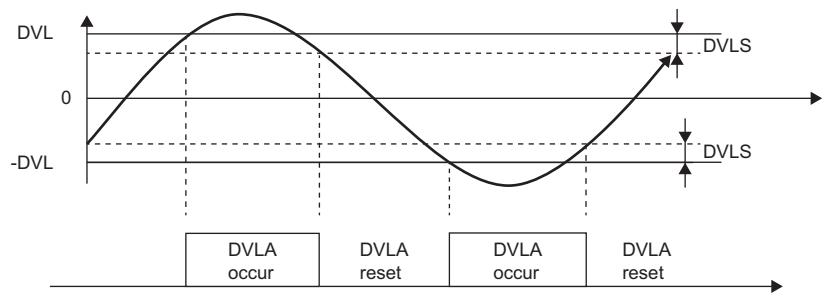
For details on tag data that is read or written with this tag access FB, refer to the following.

☞ Page 823 2PIDH

## Processing details

### Deviation check

This function block performs deviation check processing.



Condition	Alarm (ALM)
	Large deviation (DVLA)
$DVL <  DV $	TRUE (Detected)
$ DV  \leq (DVL - DVLS)$	FALSE (Reset)

DV: Deviation (%), DVLS: Large deviation alarm hysteresis (%), DVL: Deviation limit value (%)

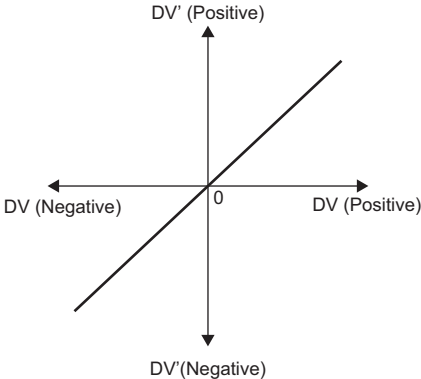
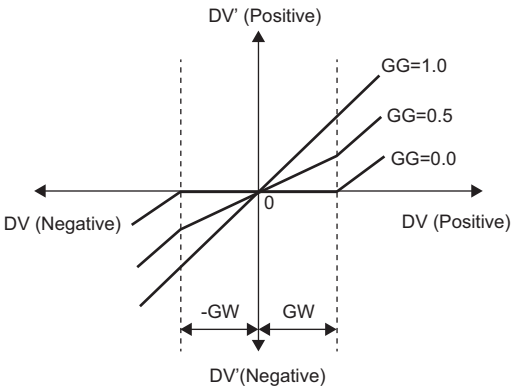
## Two-degree-of-freedom PID operation

- Gain ( $K_p$ ) is calculated as follows.

$$K_p = K \times \text{PROPORTIONAL}$$

K: Output gain, PROPORTIONAL: Gain

- Output gain (K) is calculated as follows.

Condition	Output gain (K)
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is equal to 0</li> </ul> 	$K = 1$
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is greater than 0</li> </ul> 	<p>When <math> DV  \leq GW</math></p> $K = GG$ <p>When <math> DV  &gt; GW</math></p> $K = 1 - \frac{(1-GG) \times GW}{ DV }$

DV: Deviation (%), GW: Gap width (%) = Rate of the gap width to the deviation, GG: Gap gain

- Deviation for PID operation ( $DV'$ ) is calculated as follows.

Condition	Deviation for PID operation ( $DV'$ )
$DV < -GW$	$DV' = -(GG \times GW) + (DV + GW)$
$ DV  \leq GW$	$DV' = GG \times DV$
$DV > GW$	$DV' = GG \times GW + (DV - GW)$

$DV'$ : Deviation for PID operation (%)

DV: Deviation (%)

GW: Gap width (%)

GG: Gap gain

- Deviation (DV) for direct/reverse action is calculated as follows.

Condition	Deviation (DV)
Direct action (PN = 1)	$DV (\%) = PVP (\%) - SVC (\%)$
Reverse action (PN = 0)	$DV (\%) = SVC (\%) - PVP (\%)$

DV: Deviation (%)

PVP (%): PV input value (%)

$$SVC (\%) = \frac{100}{RH-RL} \times (SVC-RL)$$

RH: Engineering value high limit

RL: Engineering value low limit

SVC: Setting value (current)

- PID operations are conducted as follows.

Item	Direct action	Reverse action
Deviation (DVn)	$DV_n = PV_n - SV_n$	$DV_n = SV_n - PV_n$
Output variation ( $\Delta MV$ )	$\Delta MV = K_p \times \left\{ \underbrace{(1 - \alpha)}_{\text{Gain}} \times \underbrace{(DV_n - DV_{n-1})}_{\text{Proportional}} + \underbrace{\frac{CT}{T_i}}_{\text{Integral}} \times DV_n \right.$ $\left. + \underbrace{(1 - \beta)}_{\text{Derivative}} \times B_n + \underbrace{\alpha \times C_n + \beta \times D_n}_{\text{Feed forward compensation}} \right\}$	
Bn	$B_n = B_{n-1} + \frac{Md \times Td}{Md \times CT + Td} \times \left\{ (DV_n - 2DV_{n-1} + DV_{n-2}) - \frac{CT \times B_{n-1}}{Td} \right\}$	
Dn	$D_n = D_{n-1} + \frac{Md \times Td}{Md \times CT + Td} \times \left\{ (PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times D_{n-1}}{Td} \right\}$	$D_n = D_{n-1} + \frac{Md \times Td}{Md \times CT + Td} \times \left\{ -(PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times D_{n-1}}{Td} \right\}$
Cn	$C_n = PV_n - PV_{n-1}$	$C_n = -(PV_n - PV_{n-1})$

- Kp: Gain
- Ti: Integral time
- Td: Derivative time
- Md: Derivative gain
- CT: Control cycle
- DVn: Deviation
- DVn-1: Last deviation value
- DVn-2: Deviation value before the last value
- PVn: Process variable
- PVn-1: Last process variable
- PVn-2: Process variable before the last value
- SVn: Engineering value conversion processing result
- $\alpha$ : Two-degree-of-freedom parameter (feedforward proportional)
- $\beta$ : Two-degree-of-freedom parameter (feedforward differential)

The integral term and derivative term are as follows under the following conditions.

Item	Condition	Processing
Derivative term	When Td = 0	Bn = 0
	When the control mode is MAN	
	When the control mode is CMV	
Integral term	When Ti = 0	$\frac{CT}{T_i} \times DV_n = 0$
	When either of MH or ML error has occurred, MVP > MH and the following expression is satisfied	
	When either of MH or ML error has occurred, MVP < ML and the following expression is satisfied	

- Ti: Integral time
- CT: Control cycle
- DVn: Deviation
- MH: Output high limit value
- ML: Output low limit value
- MVP: MV Internal operation value

Set an integral multiple of the execution cycle ( $\Delta T$ ) as a control cycle (CT).  
 Set 0.0 or a value equal to or larger than the control cycle (CT) as an integral constant.  
 PID operations of this tag access FB are performed every control cycle (CT) ( $\Delta MV$  output).  
 In other execution cycles ( $\Delta T$ ), the last value is held ( $\Delta MV = 0$ ).

## ■Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH-RL}{100} \times \text{Setting value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value (target)

## ■Inverse engineering value conversion


This function block converts SVC of an engineering value into SVC in percentage (%).

$$SVC(\%) = \frac{100}{RH-RL} \times (SVC-RL)$$

RH: Engineering value high limit, RL: Engineering value low limit, SVC: Setting value (current)

## ■Disable alarm detection

This function sets whether to detect alarms (ALM) or not in the deviation check and SV variation rate & high/low limiter processing.


Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) and the disable alarm detection 2 (INH2) are TRUE, DVLA of the alarm (ALM), or DSVLA, SVHA, or SVLA of the alarm 2 (ALM2) will not be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DVLI</li> <li>• DSVLI</li> <li>• SVHI</li> <li>• SVLI</li> </ul>
Disable alarm detection by control mode selection	No alarm will be detected when the control mode is CASDR.
Disable alarm detection by loop stop processing	 Page 268 Loop stop processing

## ■Auto tuning (AT1, AT2)

This function block detects dynamic characteristics and automatically calculates a proportional gain (Kp), integral time (Ti), and derivative time (Td) for PID operations.


Select the step response method or the limit cycle method for auto tuning.

- AT1 (Step response method)

Auto tuning is for setting initial values of the proportional gain (Kp), integral time (Ti), and derivative time (Td) for PID operations. The ZN method: Ziegler-Nichols' step response method is used for calculating the values. ( Page 917 Step response method)

This method can be used in the MAN or CMV mode.

- AT2 (Limit cycle method)

In this method, a proportional gain (Kp), an integral time (Ti), and a derivative time (Td) for PID operations are calculated from the variation amplitude and variation cycle of a process variable obtained by repeatedly outputting the high and low limit values of a manipulated value. ( Page 920 Limit cycle method)

This method can be used in the MAN, AUT, CAS, CMV, or CSV mode.

## ■PV tracking function

To avoid sudden changes of the manipulated value at mode switching (from MAN to AUT), this function block matches the setting value (target) with the process variable when the control mode is MAN or CMV and keeps the value.

Condition	PV tracking processing
When PVTRK_EN is TRUE and the control mode is "MAN (CMV)"	Setting value (target) = Process variable Setting value (current) = Process variable
When PVTRK_EN is FALSE and the control mode is other than "MAN (CMV)"	Non-processing

### ■SV variation rate & high/low limiter

This function block checks the variation rate and high/low limits to the setting value every control cycle (CT).

- Variation rate limiter

When the control mode is AUT, CAS, or CSV

The SV variation rate high limit value input in percentage is converted into an engineering value, and the processing will be performed.

DSVL → DSVLT (DSVL: SV variation rate high limit value, DSVLT: SV variation rate high limit value converted into an engineering value)

Condition	Variation rate limiter result	Alarm 2 (ALM2) target variation rate limit (DSVLA)
$ SV - SVC  \leq DSVLT$	SV	FALSE (Reset)
$SV - SVC > DSVLT$	$SVC + DSVLT$	TRUE (Detected)
$SV - SVC < -DSVLT$	$SVC - DSVLT$	TRUE (Detected)

SV: Setting value (target), SVC: Setting value (current)

If DSVLI of the disable alarm detection 2 or ERRI of the disable alarm detection is TRUE, DSVLA will be FALSE.

When the control mode is MAN, CMV, or CASDR

Condition	Variation rate limiter result	Alarm 2 (ALM2) target variation rate limit (DSVLA)
None	SV	FALSE (Reset)

- High/low limiter

When the control mode is MAN, AUT, CAS, CMV, or CSV and SVLMT\_EN is TRUE

Condition	High/low limiter result	Alarm 2 (ALM2)	
		SV low limit (SVLA)	SV high limit (SVHA)
Variation rate limiter result > SH	SH	FALSE (Reset)	TRUE (Detected)
Variation rate limiter result < SL	SL	TRUE (Detected)	FALSE (Reset)
$SL \leq \text{Variation rate limiter result} \leq SH$	Variation rate limiter result	FALSE (Reset)	FALSE (Reset)

If SVLI of the disable alarm detection 2 or ERRI of the disable alarm detection is TRUE, SVLA will be FALSE.

If SVHI of the disable alarm detection 2 or ERRI of the disable alarm detection is TRUE, SVHA will be FALSE.

The high/low limiter result is stored in SVC (setting value (current)).

When the control mode is CASDR, or when SVLMT\_EN is FALSE

The variation rate limiter result is stored in SVC (set value (current)).

### ■Integration stop

This function block stops an integral element operation.

Condition	Processing
ISTP = TRUE	The integral element operation is stopped.
ISTP = FALSE	Non-processing

### ■Derivative stop

This function block stops a derivative element operation.

Condition	Processing
DSTP = TRUE	The derivative element operation is stopped.
DSTP = FALSE	Non-processing

### ■Integration stop at MV variation rate limiter occurrence

This function block stops an integral element operation when an MV variation rate limiter alarm occurs.

Condition	Processing
When LMT_ISTP is TRUE and a DMLA alarm has occurred	When the variation rate of a manipulated value has exceeded a positive/negative limit value and the sign (positive/negative) of an integral operation value is the same as that of the limit value, the integral element operation is stopped.
LMT_ISTP = FALSE	Non-processing

## Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) or the tag stop (TSTP) of the monitor output buffer (DOM) is TRUE, this function block performs the following processing.

- $\Delta MV$  is cleared to 0.
- The control mode is automatically switched to MANUAL.
- When DVLA of the alarm (ALM) has occurred, the DVLA is reset. When DSVLA, SVLA, or SVHA of the alarm 2 (ALM2) has occurred, the DSVLA, SVLA, or SVHA is reset.
- No alarm is detected in the deviation check or SV variation rate & high/low limiter processing.

## Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation					
	Deviation check	Two-degree-of-freedom PID operation	Engineering value conversion	Inverse engineering value conversion	Alarm	Auto tuning (AT1)
MAN, CMV	○	○	×	○	○ <sup>*1</sup>	○
AUT	○	○	×	○	○ <sup>*1</sup>	×
CAS, CSV	○	○	○	○	○ <sup>*1</sup>	×
CASDR	×	○	○	○	×	×

Control mode	Processing operation				
	Auto tuning (AT2)	PV tracking	SV variation rate & high/low limiter	Integration stop	Derivative stop
MAN, CMV	○	○	○ <sup>*2,3</sup>	○	○
AUT	○	×	○ <sup>*3</sup>	○	○
CAS, CSV	○	×	○ <sup>*3</sup>	○	○
CASDR	×	×	×	○	○

\*1 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

\*2 In the MAN mode, SV variation rate limiter processing is not performed.

\*3 This processing is not performed when a sensor error (SEA) has occurred and output hold has been selected. SVC (setting value (current)) is also not updated.

## Operation error

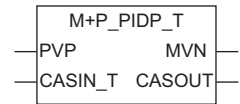
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data, operation constant, tag data, or operation result within a function block is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The sampling interval time for AT1 (AT1ST) setting is less than 0.
	The time-out interval for AT1 (AT1TOUT1) setting is less than 0.
	The time-out interval after maximum slope for AT1 (AT1TOUT2) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the sampling interval time for AT1 (AT1ST) by the execution cycle ( $\Delta T$ ) exceeds 32767.
	The value divided the time-out interval for AT1 (AT1TOUT1) by the execution cycle ( $\Delta T$ ) exceeds 32767.
	The value divided the time-out interval after maximum slope for AT1 (AT1TOUT2) by the execution cycle ( $\Delta T$ ) exceeds 32767.
	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.	

# 15.9 Position Type PID Control (Enable Tracking for primary loop/Disable Tracking from secondary loop) (M+P\_PIDP\_T)

## M+P\_PIDP\_T

This FB performs PID operations using process variable differential, inexact differential, and position type, and outputs the result. The primary loop can be tracked.

### FBD/LD



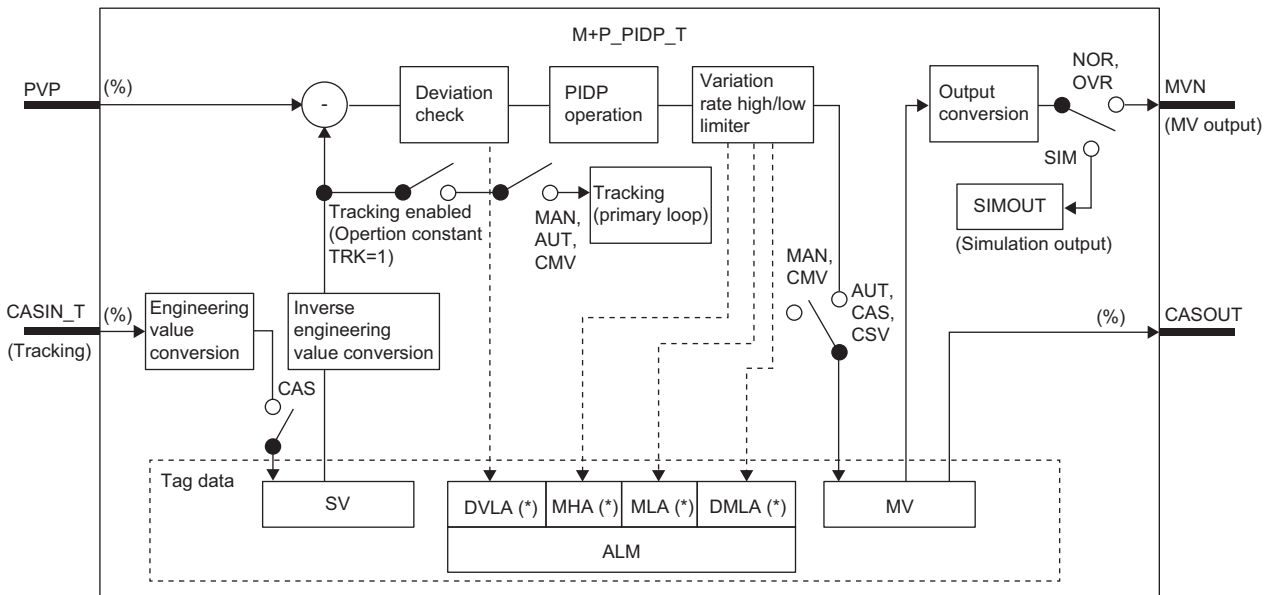
### Applicable tag type

PIDP

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bits item.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN_T	Cascade SV input (unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVN	MV output	NMIN to NMAX	Output variable	REAL
CASOUT	Cascade MV output (unit: %)	0 to 100[%]	Output variable	REAL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MTD	Derivative Gain	0 to 9999	8.0	User	REAL
DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
SVPTN_B0	Setting value (SV) used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SVPTN_B1	Setting value (SV) pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

## Public variable (others) <sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

\*1 Read or write the variables using a program. They are not displayed in "FB Property" of the engineering tool.

## Tag data

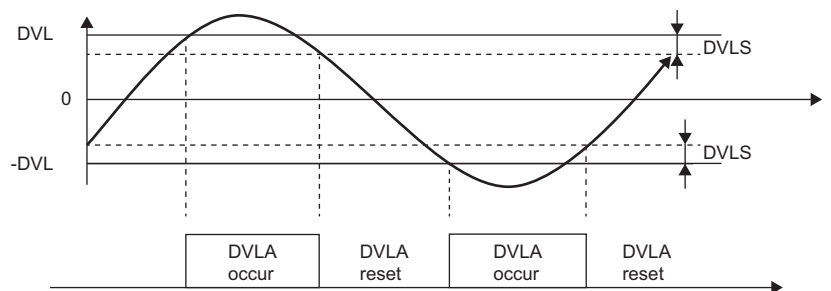
For details on tag data that is read or written with this tag access FB, refer to the following.

Page 829 PIDP

## Processing details

### Deviation check

This function block performs deviation check processing.



Condition	Alarm (ALM)
	Large deviation (DVLA)
$DVL <  DV $	TRUE (Detected)
$ DV  \leq (DVL - DVLS)$	FALSE (Reset)

DV: Deviation (%), DVLS: Large deviation alarm hysteresis (%), DVL: Deviation limit value (%)



## ■PIDP operation

- Gain ( $K_p$ ) is calculated as follows.

$$K_p = K \times \text{PROPORTIONAL}$$

K: Output gain, PROPORTIONAL: Gain

- Output gain (K) is calculated as follows.

Condition	Output gain (K)
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is equal to 0</li> </ul>	$K = 1$
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is greater than 0</li> </ul>	<p>When <math> DV  \leq GW</math></p> $K = GG$
	<p>When <math> DV  &gt; GW</math></p> $K = 1 - \frac{(1-GG) \times GW}{ DV }$

DV: Deviation (%), GW: Gap width (%) = Rate of the gap width to the deviation, GG: Gap gain

- Deviation for PIDP operation ( $DV'$ ) is calculated as follows.

Condition	Deviation for PIDP operation ( $DV'$ )
$DV < -GW$	$DV' = -(GG \times GW) + (DV + GW)$
$ DV  \leq GW$	$DV' = GG \times DV$
$DV > GW$	$DV' = GG \times GW + (DV - GW)$

$DV'$ : Deviation for PIDP operation (%)

DV: Deviation (%)

GW: Gap width (%)

GG: Gap gain

- Deviation (DV) for direct/reverse action is calculated as follows.

Condition	Deviation (DV)
Direct action (PN = 1)	$DV (\%) = PVP (\%) - SV (\%)$
Reverse action (PN = 0)	$DV (\%) = SV (\%) - PVP (\%)$

DV: Deviation (%)

PVP (%): PV input value (%)

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit

RL: Engineering value low limit

SV: Setting value

- PIDP operations are conducted as follows.

Item	Direct action	Reverse action
Deviation (DVn)	$DV_n = PV_n - SV_n$	$DV_n = SV_n - PV_n$
Output variation (MV)	$MV = K_p \times \{ \underbrace{DV_n}_{\text{Proportional}} + \underbrace{I_n}_{\text{Integral}} + \underbrace{B_n}_{\text{Derivative}} \}$	
In	$I_n = I_{n-1} + \frac{CT}{Ti} \times DV_n$	
Bn	$B_n = B_{n-1} + \frac{Md \times Td}{Md \times CT + Td} \times \{ (PV_n - PV_{n-1}) - \frac{CT \times B_{n-1}}{Td} \}$	$B_n = B_{n-1} + \frac{Md \times Td}{Md \times CT + Td} \times \{ -(PV_n - PV_{n-1}) - \frac{CT \times B_{n-1}}{Td} \}$

Kp: Gain  
 Ti: Integral time  
 Td: Derivative time  
 Md: Derivative gain  
 CT: Control cycle  
 DVn: Deviation  
 DVn-1: Last deviation value  
 PVn: Process variable  
 PVn-1: Last process variable  
 SVn: Engineering value conversion processing result

The integral term and derivative term are as follows under the following conditions.

Item	Condition	Processing
Derivative term	When $Td = 0$	$B_n = 0$
	When the control mode is MAN	
	When the control mode is CMV	
Integral term	When $Ti = 0$	$\frac{CT}{Ti} \times DV_n = 0$
	When an MH error has occurred and the following expression is satisfied $\frac{CT}{Ti} \times DV_n > 0$	
	When an ML error has occurred and the following expression is satisfied $\frac{CT}{Ti} \times DV_n < 0$	

Ti: Integral time  
 CT: Control cycle  
 DVn: Deviation  
 MH: Output high limit value  
 ML: Output low limit value

Set an integral multiple of the execution cycle ( $\Delta T$ ) as a control cycle (CT).

Set 0.0 or a value equal to or larger than the control cycle (CT) as an integral constant.

PIDP operations of this tag access FB are performed every control cycle (CT) (MV output).

In other execution cycles ( $\Delta T$ ), the last manipulated value is held.

### ■Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH - RL}{100} \times \text{Setting value (\%)} \text{ from the primary loop} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Inverse engineering value conversion

This function block converts the setting value (SV) of an engineering value into a setting value (SV) in percentage (%).

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Tracking processing

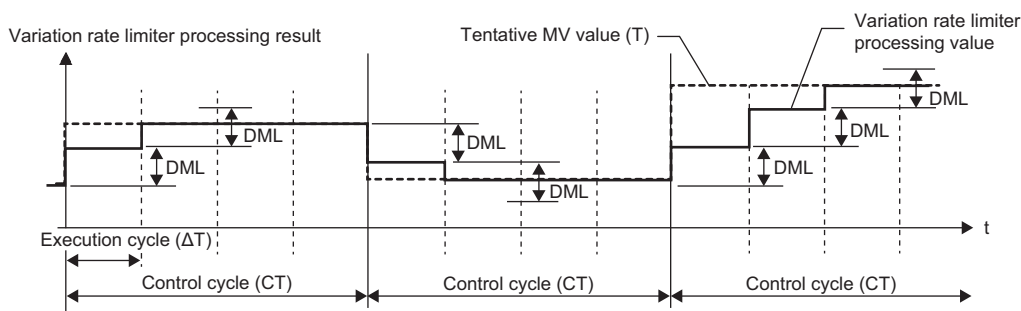
The following table shows whether tracking processing to the input variable CASIN\_T is performed or not.

Condition		Result
Tracking Flag (TRK)	Setting Value (SV) Used (SVPTN_B0)	
1	FALSE	The input variable CASIN_T is tracked.
	TRUE	The input variable CASIN_T is not tracked.
0	FALSE or TRUE	

## ■Variation rate & high/low limiter

This function block checks the variation rate and high/low limits of the input value.

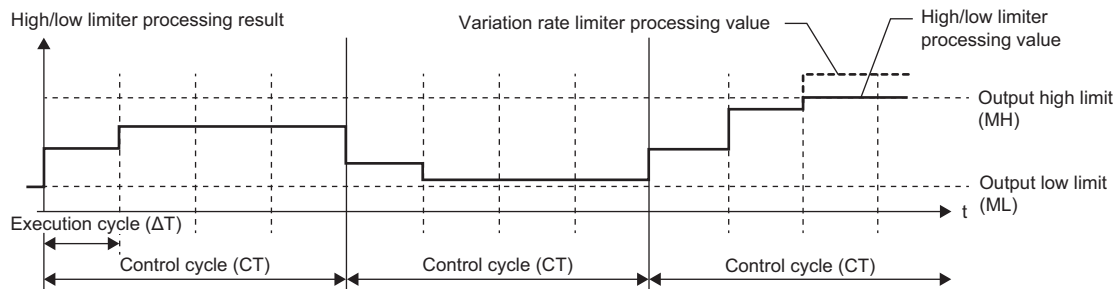
- Variation rate limiter



Condition	Variation rate limiter processing result	Alarm (ALM)
		Output variation rate limit (DMLA)
$ T - MV  \leq DML$	T	FALSE (Reset)
$T - MV > DML$	$MV + DML$	TRUE (Detected)
$T - MV < -DML$	$MV - DML$	TRUE (Detected)

T: Tentative manipulated value, MV: Manipulated value, DML: Output variation rate high limit value

- High/low limiter

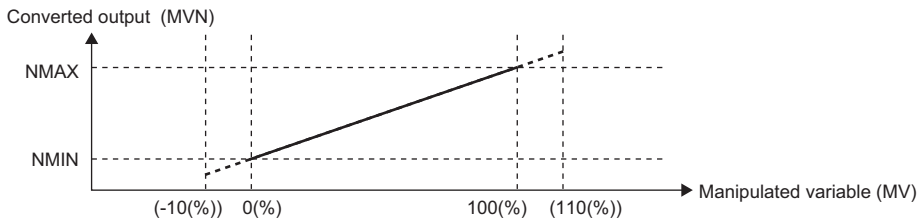


Condition	High/low limiter processing result	Alarm (ALM)	
		Output Low Limit Alarm (MLA)	Output High Limit Alarm (MHA)
Variation rate limiter processing result > MH	MH	FALSE (Reset)	TRUE (Detected)
Variation rate limiter processing result < ML	ML	TRUE (Detected)	FALSE (Reset)
$ML \leq$ Variation rate limiter processing result $\leq$ MH	Variation rate limiter processing value	FALSE (Reset)	FALSE (Reset)

MH: Output high limit value, ML: Output low limit value

## Output conversion

This function block performs output conversion processing.



$$\text{Converted output (MVN)} = \left\{ (\text{NMAX} - \text{NMIN}) \times \frac{\text{MV}}{100} \right\} + \text{NMIN}$$

NMAX: Output conversion high limit value

NMIN: Output conversion low limit

MV: Manipulated value (%)

MVN: Output conversion output value

## Disable alarm detection

This function sets whether to detect alarms (ALM) or not in the deviation check and variation rate & high/low limiter processing.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, DVLA, DMLA, MHA, or MLA of the alarm (ALM) will not be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DVLI</li> <li>• DMLI</li> <li>• MHI</li> <li>• MLI</li> </ul>
Disable alarm detection by control mode selection	In the MAN or CMV mode, DVLA, DMLA, MHA, and MLA of the alarm (ALM) are reset and the DVLA, DMLA, MHA, or MLA will not be detected.
Disable alarm detection by loop stop processing	Page 274 Loop stop processing

## Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- Outputs (MVN) are held.
- The control mode is automatically switched to MANUAL.
- When DVLA, DMLA, MHA, or MLA of the alarm (ALM) has occurred, the DVLA, DMLA, MHA, or MLA is reset.
- No alarm is detected in the variation rate & high/low limiter processing.

## Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation							
	Deviation check	PIDP operation	Engineering value conversion	Inverse engineering value conversion	Variation rate & high/low limiter	Output conversion	Tracking	Alarm
MAN, CMV	○	○	×	○	×	○	○ <sup>*1</sup>	○ <sup>*2</sup>
AUT	○	○	×	○	○	○	○ <sup>*1</sup>	○ <sup>*2</sup>
CAS, CSV	○	○	○	○	○	○	×	○ <sup>*2</sup>

\*1 Tracking is performed when the tracking flag (TRK) is 1.

\*2 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

## Operation error

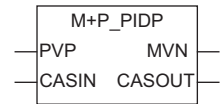
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP), operation constant (MTD, DVLS, PN, TRK, SVPTN_B0, SVPTN_B1, NMAX, NMIN), or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.

# 15.10 Position Type PID Control (Disable Tracking for primary loop/Disable Tracking from secondary loop) (M+P\_PIDP)

## M+P\_PIDP

This FB performs PID operations using process variable differential, inexact differential, and position type, and outputs the result.

### FBD/LD



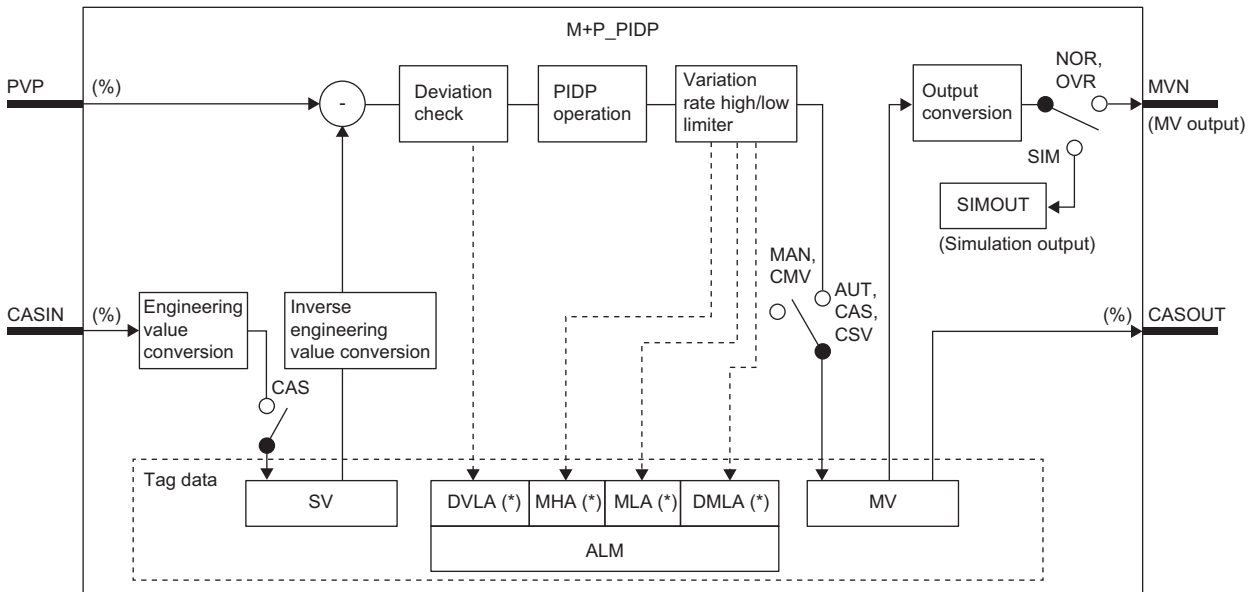
### Applicable tag type

PIDP

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bits item.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN	Cascade SV input (unit: %)	0 to 100[%]	Input variable	REAL
MVN	MV output	NMIN to NMAX	Output variable	REAL
CASOUT	Cascade MV output (unit: %)	0 to 100[%]	Output variable	REAL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MTD	Derivative Gain	0 to 9999	8.0	User	REAL
DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
SVPTN_B0	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL

## Public variable (others) \*1

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

\*1 Read or write the variables using a program. They are not displayed in "FB Property" of the engineering tool.

## Tag data

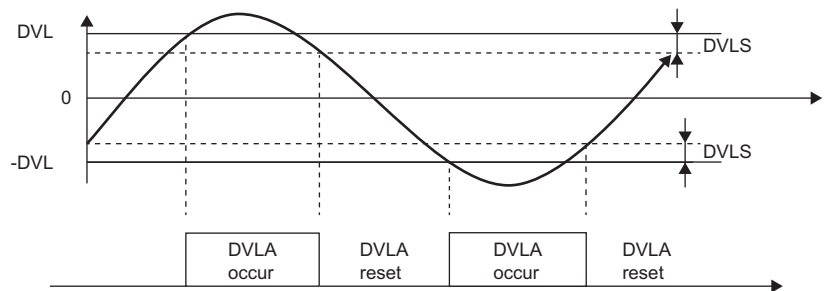
For details on tag data that is read or written with this tag access FB, refer to the following.

Page 829 PIDP

## Processing details

### Deviation check

This function block performs deviation check processing.



Condition	Alarm (ALM)
	Large deviation (DVLA)
$DVL <  DV $	TRUE (Detected)
$ DV  \leq (DVL - DVLS)$	FALSE (Reset)

DV: Deviation (%), DVLS: Large deviation alarm hysteresis (%), DVL: Deviation limit value (%)

## ■PIDP operation

- Gain ( $K_p$ ) is calculated as follows.

$$K_p = K \times \text{PROPORTIONAL}$$

K: Output gain, PROPORTIONAL: Gain

- Output gain (K) is calculated as follows.

Condition	Output gain (K)
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is equal to 0</li> </ul>	$K = 1$
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is greater than 0</li> </ul>	<p>When <math> DV  \leq GW</math></p> $K = GG$
	<p>When <math> DV  &gt; GW</math></p> $K = 1 - \frac{(1-GG) \times GW}{ DV }$

DV: Deviation (%), GW: Gap width (%) = Rate of the gap width to the deviation, GG: Gap gain

- Deviation for PIDP operation ( $DV'$ ) is calculated as follows.

Condition	Deviation for PIDP operation ( $DV'$ )
$DV < -GW$	$DV' = -(GG \times GW) + (DV + GW)$
$ DV  \leq GW$	$DV' = GG \times DV$
$DV > GW$	$DV' = GG \times GW + (DV - GW)$

$DV'$ : Deviation for PIDP operation (%)

DV: Deviation (%)

GW: Gap width (%)

GG: Gap gain

- Deviation (DV) for direct/reverse action is calculated as follows.

Condition	Deviation (DV)
Direct action (PN = 1)	$DV (\%) = PVP (\%) - SV (\%)$
Reverse action (PN = 0)	$DV (\%) = SV (\%) - PVP (\%)$

DV: Deviation (%)

PVP (%): PV input value (%)

$$SV (\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit

RL: Engineering value low limit

SV: Setting value



- PIDP operations are conducted as follows.

Item	Direct action	Reverse action
Deviation (DVn)	$DV_n = PV_n - SV_n$	$DV_n = SV_n - PV_n$
Output variation (MV)	$MV = K_p \times \left\{ \underbrace{DV_n}_{\text{Gain}} + \underbrace{I_n}_{\text{Integral}} + \underbrace{B_n}_{\text{Derivative}} \right\}$	
I <sub>n</sub>	$I_n = I_{n-1} + \frac{CT}{Ti} \times DV_n$	
B <sub>n</sub>	$B_n = B_{n-1} + \frac{Md \times Td}{Md \times CT + Td} \times \left\{ (PV_n - PV_{n-1}) - \frac{CT \times B_{n-1}}{Td} \right\}$	$B_n = B_{n-1} + \frac{Md \times Td}{Md \times CT + Td} \times \left\{ -(PV_n - PV_{n-1}) - \frac{CT \times B_{n-1}}{Td} \right\}$

- Kp: Gain
- Ti: Integral time
- Td: Derivative time
- Md: Derivative gain
- CT: Control cycle
- DVn: Deviation
- DVn-1: Last deviation value
- PVn: Process variable
- PVn-1: Last process variable
- SVn: Engineering value conversion processing result

The integral term and derivative term are as follows under the following conditions.

Item	Condition	Processing
Derivative term	When Td = 0	B <sub>n</sub> = 0
	When the control mode is MAN	
	When the control mode is CMV	
Integral term	When Ti = 0	$\frac{CT}{Ti} \times DV_n = 0$
	When an MH error has occurred and the following expression is satisfied	
	When an ML error has occurred and the following expression is satisfied	

- Ti: Integral time
- CT: Control cycle
- DVn: Deviation
- MH: Output high limit value
- ML: Output low limit value

Set an integral multiple of the execution cycle (ΔT) as a control cycle (CT).

Set 0.0 or a value equal to or larger than the control cycle (CT) as an integral constant.

PIDP operations of this tag access FB are performed every control cycle (CT) (MV output).

In other execution cycles (ΔT), the last manipulated value is held.

## ■Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH-RL}{100} \times \text{Setting value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Inverse engineering value conversion

This function block converts the setting value (SV) of an engineering value into a setting value (SV) in percentage (%).

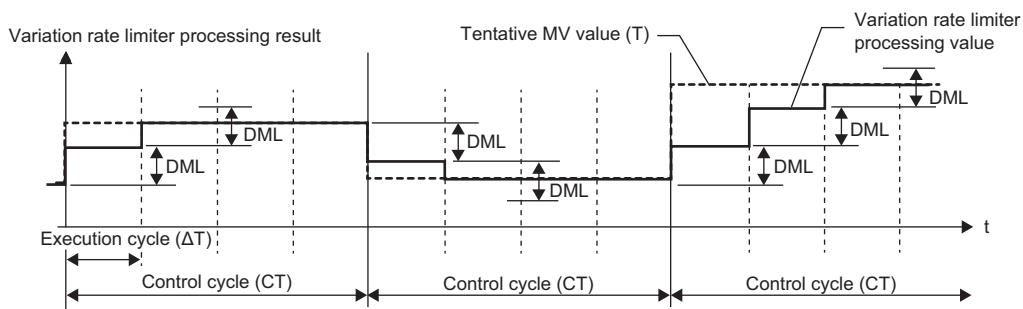
$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Variation rate & high/low limiter

This function block checks the variation rate and high/low limits of the input value.

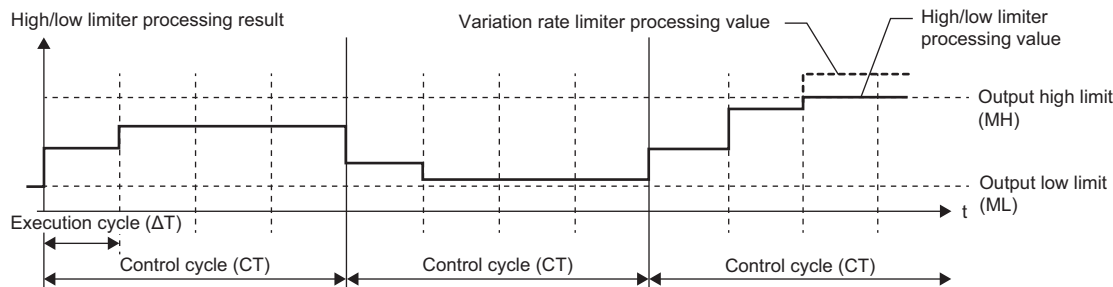
- Variation rate limiter



Condition	Variation rate limiter processing result	Alarm (ALM)
		Output variation rate limit (DMLA)
$ T - MV  \leq DML$	T	FALSE (Reset)
$T - MV > DML$	$MV + DML$	TRUE (Detected)
$T - MV < -DML$	$MV - DML$	TRUE (Detected)

T: Tentative manipulated value, MV: Manipulated value, DML: Output variation rate high limit value

- High/low limiter

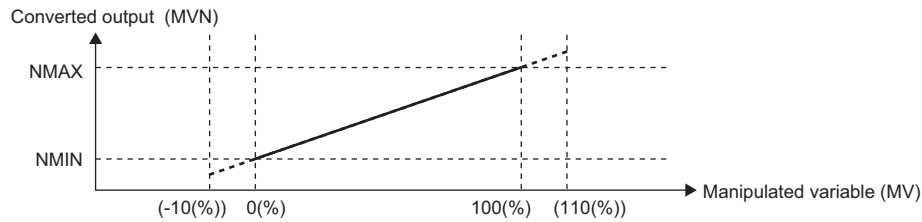


Condition	High/low limiter processing result	Alarm (ALM)	
		Output Low Limit Alarm (MLA)	Output High Limit Alarm (MHA)
Variation rate limiter processing result > MH	MH	FALSE (Reset)	TRUE (Detected)
Variation rate limiter processing result < ML	ML	TRUE (Detected)	FALSE (Reset)
$ML \leq \text{Variation rate limiter processing result} \leq MH$	Variation rate limiter processing value	FALSE (Reset)	FALSE (Reset)

MH: Output high limit value, ML: Output low limit value

## Output conversion

This function block performs output conversion processing.



$$\text{Converted output (MVN)} = \left\{ (\text{NMAX} - \text{NMIN}) \times \frac{\text{MV}}{100} \right\} + \text{NMIN}$$

NMAX: Output conversion high limit value

NMIN: Output conversion low limit

MV: Manipulated value (%)

MVN: Output conversion output value

## Disable alarm detection

This function sets whether to detect alarms (ALM) or not in the deviation check and variation rate & high/low limiter processing.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, DVLA, DMLA, MHA, or MLA of the alarm (ALM) will not be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DVLI</li> <li>• DMLI</li> <li>• MHI</li> <li>• MLI</li> </ul>
Disable alarm detection by control mode selection	In the MAN or CMV mode, DVLA, DMLA, MHA, and MLA of the alarm (ALM) are reset and the DVLA, DMLA, MHA, or MLA will not be detected.
Disable alarm detection by loop stop processing	Page 281 Loop stop processing

## Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- Outputs (MVN) are held.
- The control mode is automatically switched to MANUAL.
- When DVLA, DMLA, MHA, or MLA of the alarm (ALM) has occurred, the DVLA, DMLA, MHA, or MLA is reset.
- No alarm is detected in the variation rate & high/low limiter processing.

## Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation						
	Deviation check	PIDP operation	Engineering value conversion	Inverse engineering value conversion	Variation rate & high/low limiter	Output conversion	Alarm
MAN, CMV	○	○	×	○	×	○	○*1
AUT	○	○	×	○	○	○	○*1
CAS, CSV	○	○	○	○	○	○	○*1

\*1 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP), operation constant (MTD, DVLS, PN, SVPTN_B0, NMAX, NMIN), or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.

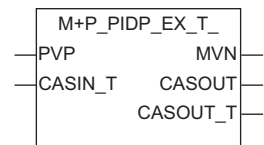
# 15.11 Position Type PID Control (Enable Tracking for primary loop/Enable Tracking from secondary loop) (M+P\_PIDP\_EX\_T\_)

## M+P\_PIDP\_EX\_T\_

This FB performs PID operations using process variable differential, inexact differential, and position type, and outputs the result.

It also allows manipulated value bumpless switching and tracking from the primary loop and secondary loop at a change of control mode.

### FBD/LD



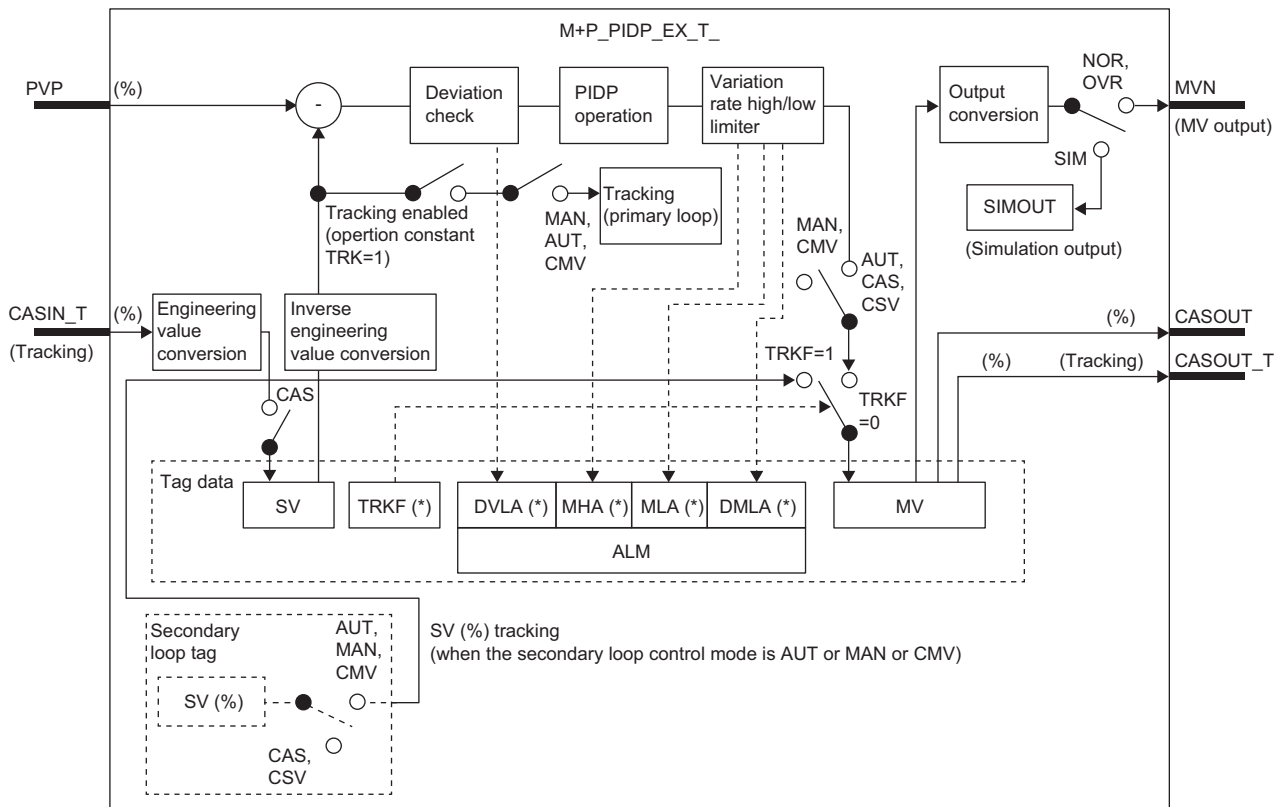
### Applicable tag type

PIDP

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bits item.

## Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN_T	Cascade SV input (unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVN	MV output	NMIN to NMAX	Output variable	REAL
CASOUT	Cascade MV output (unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MTD	Derivative Gain	0 to 9999	8.0	User	REAL
DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
SVPTN_B0	Setting value (SV) used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SVPTN_B1	Setting value (SV) pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

### ■Public variable (others) <sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

\*1 Read or write the variables using a program. They are not displayed in "FB Property" of the engineering tool.

### ■Tag data

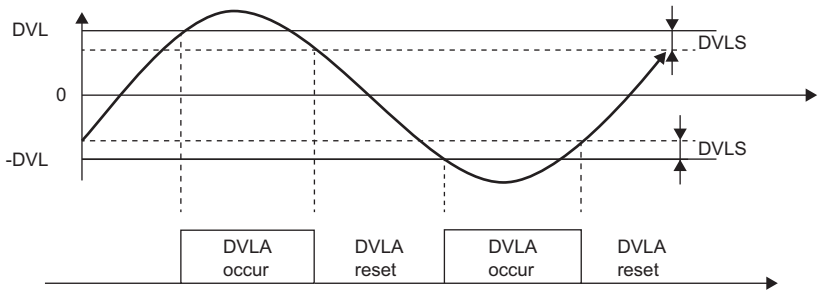
For details on tag data that is read or written with this tag access FB, refer to the following.

☞ Page 829 PIDP

## Processing details

### Deviation check

This function block performs deviation check processing.



Condition	Alarm (ALM)
	Large deviation (DVLA)
$DVL <  DV $	TRUE (Detected)
$ DV  \leq (DVL - DVLS)$	FALSE (Reset)

DV: Deviation (%), DVLS: Large deviation alarm hysteresis (%), DVL: Deviation limit value (%)

### PIDP operation

- Gain ( $K_p$ ) is calculated as follows.

$$K_p = K \times \text{PROPORTIONAL}$$

K: Output gain, PROPORTIONAL: Gain

- Output gain (K) is calculated as follows.

Condition	Output gain (K)
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is equal to 0</li> </ul>	$K = 1$
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is greater than 0</li> </ul>	When $ DV  \leq GW$ $K = GG$
	When $ DV  > GW$ $K = 1 - \frac{(1-GG) \times GW}{ DV }$

DV: Deviation (%), GW: Gap width (%) = Rate of the gap width to the deviation, GG: Gap gain

- Deviation for PIDP operation (DV') is calculated as follows.

Condition	Deviation for PIDP operation (DV')
DV < -GW	DV' = -(GG × GW) + (DV + GW)
DV  ≤ GW	DV' = GG × DV
DV > GW	DV' = GG × GW + (DV - GW)

DV': Deviation for PIDP operation (%)  
 DV: Deviation (%)  
 GW: Gap width (%)  
 GG: Gap gain

- Deviation (DV) for direct/reverse action is calculated as follows.

Condition	Deviation (DV)
Direct action (PN = 1)	DV (%) = PVP (%) - SV (%)
Reverse action (PN = 0)	DV (%) = SV (%) - PVP (%)

DV: Deviation (%)  
 PVP (%): PV input value (%)

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit  
 RL: Engineering value low limit  
 SV: Setting value

- PIDP operations are conducted as follows.

Item	Direct action	Reverse action
Deviation (DVn)	DVn = PVn - SVn	DVn = SVn - PVn
Output variation (MV)	$MV = K_p \times \left\{ \underbrace{DV_n}_{\text{Proportional}} + \underbrace{I_n}_{\text{Integral}} + \underbrace{B_n}_{\text{Derivative}} \right\}$	
In	$I_n = I_{n-1} + \frac{CT}{Ti} \times DV_n$	
Bn	$B_n = B_{n-1} + \frac{Md \times Td}{Md \times CT + Td} \times \left\{ (PV_n - PV_{n-1}) - \frac{CT \times B_{n-1}}{Td} \right\}$	$B_n = B_{n-1} + \frac{Md \times Td}{Md \times CT + Td} \times \left\{ -(PV_n - PV_{n-1}) - \frac{CT \times B_{n-1}}{Td} \right\}$

Kp: Gain  
 Ti: Integral time  
 Td: Derivative time  
 Md: Derivative gain  
 CT: Control cycle  
 DVn: Deviation  
 DVn-1: Last deviation value  
 PVn: Process variable  
 PVn-1: Last process variable  
 SVn: Engineering value conversion processing result



The integral term and derivative term are as follows under the following conditions.

Item	Condition	Processing
Derivative term	When $T_d = 0$	$B_n = 0$
	When the control mode is MAN	
	When the control mode is CMV	
Integral term	When $T_i = 0$	$\frac{CT}{T_i} \times DV_n = 0$
	When an MH error has occurred and the following expression is satisfied $\frac{CT}{T_i} \times DV_n > 0$	
	When an ML error has occurred and the following expression is satisfied $\frac{CT}{T_i} \times DV_n < 0$	
	When the control mode is MAN	
	When the control mode is CMV	
	When tracking from the secondary loop is performed	$I_{n-1} = \frac{MV}{K_p} - (DV_n + B_n)$
	When the control mode is changed from MAN or CMV to AUT, CAS, or CSV	

Ti: Integral time  
 CT: Control cycle  
 DVn: Deviation  
 MH: Output high limit value  
 ML: Output low limit value

Set an integral multiple of the execution cycle ( $\Delta T$ ) as a control cycle (CT).

Set 0.0 or a value equal to or larger than the control cycle (CT) as an integral constant.

PIDP operations of this tag access FB are performed every control cycle (CT) (MV output).

In other execution cycles ( $\Delta T$ ), the last manipulated value is held.

### ■ Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH-RL}{100} \times \text{Setting value (\%)} \text{ from the primary loop} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Inverse engineering value conversion

This function block converts the setting value (SV) of an engineering value into a setting value (SV) in percentage (%).

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Tracking processing

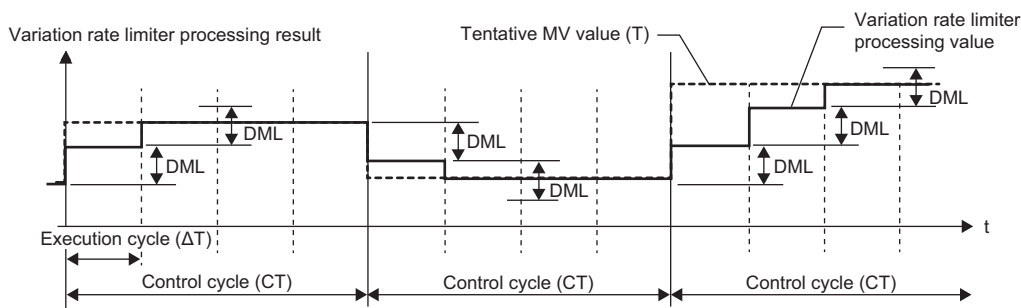
The following table shows whether tracking processing to the input variable CASIN\_T is performed or not.

Condition		Result
Tracking Flag (TRK)	Setting Value (SV) Used (SVPTN_B0)	
1	FALSE	The input variable CASIN_T is tracked.
	TRUE	The input variable CASIN_T is not tracked.
0	FALSE or TRUE	

## ■Variation rate & high/low limiter

This function block checks the variation rate and high/low limits of the input value.

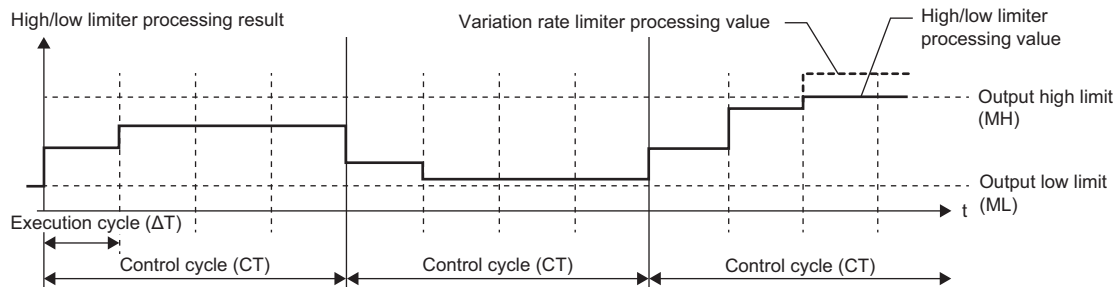
- Variation rate limiter



Condition	Variation rate limiter processing result	Alarm (ALM)
		Output variation rate limit (DMLA)
$ T - MV  \leq DML$	T	FALSE (Reset)
$T - MV > DML$	$MV + DML$	TRUE (Detected)
$T - MV < -DML$	$MV - DML$	TRUE (Detected)

T: Tentative manipulated value, MV: Manipulated value, DML: Output variation rate high limit value

- High/low limiter

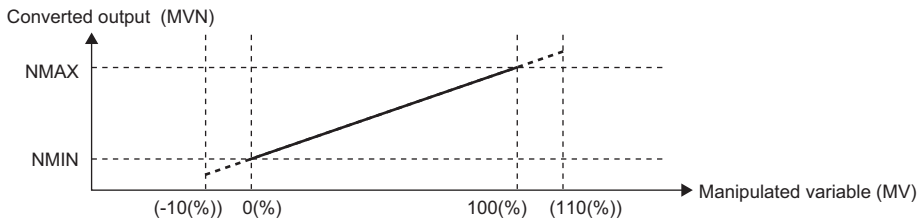


Condition	High/low limiter processing result	Alarm (ALM)	
		Output Low Limit Alarm (MLA)	Output High Limit Alarm (MHA)
Variation rate limiter processing result > MH	MH	FALSE (Reset)	TRUE (Detected)
Variation rate limiter processing result < ML	ML	TRUE (Detected)	FALSE (Reset)
$ML \leq \text{Variation rate limiter processing result} \leq MH$	Variation rate limiter processing value	FALSE (Reset)	FALSE (Reset)

MH: Output high limit value, ML: Output low limit value

## Output conversion

This function block performs output conversion processing.



$$\text{Converted output (MVN)} = \left\{ (\text{NMAX} - \text{NMIN}) \times \frac{\text{MV}}{100} \right\} + \text{NMIN}$$

NMAX: Output conversion high limit value

NMIN: Output conversion low limit

MV: Manipulated value (%)

MVN: Output conversion output value

## Disable alarm detection

This function sets whether to detect alarms (ALM) or not in the deviation check and variation rate & high/low limiter processing.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, DVLA, DMLA, MHA, or MLA of the alarm (ALM) will not be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DVLI</li> <li>• DMLI</li> <li>• MHI</li> <li>• MLI</li> </ul>
Disable alarm detection by control mode selection	In the MAN or CMV mode, DVLA, DMLA, MHA, and MLA of the alarm (ALM) are reset and the DVLA, DMLA, MHA, or MLA will not be detected.
Disable alarm detection by loop stop processing	Page 289 Loop stop processing

## Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- Outputs (MVN) are held.
- The control mode is automatically switched to MANUAL.
- When DVLA, DMLA, MHA, or MLA of the alarm (ALM) has occurred, the DVLA, DMLA, MHA, or MLA is reset.
- No alarm is detected in the variation rate & high/low limiter processing.

## Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation							
	Deviation check	PIDP operation	Engineering value conversion	Inverse engineering value conversion	Variation rate & high/low limiter	Output conversion	Tracking	Alarm
MAN, CMV	○	○	×	○	×	○	○ <sup>*1</sup>	○ <sup>*2</sup>
AUT	○	○	×	○	○	○	○ <sup>*1</sup>	○ <sup>*3</sup>
CAS, CSV	○	○	○	○	○	○	×	○ <sup>*3</sup>

\*1 Tracking is performed when the tracking flag (TRK) is 1.

\*2 An alarm (ALM) whose corresponding bit is TRUE (detected) is reset, and the alarm will not be detected.

\*3 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP), operation constant (MTD, DVLS, PN, TRK, SVPTN_B0, SVPTN_B1, NMAX, NMIN), or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.

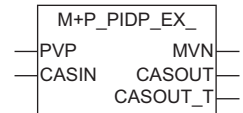
# 15.12 Position Type PID Control (Disable Tracking for primary loop/Enable Tracking from secondary loop) (M+P\_PIDP\_EX\_)

## M+P\_PIDP\_EX\_

This FB performs PID operations using process variable differential, inexact differential, and position type, and outputs the result.

It also allows manipulated value bumpless switching and tracking from the secondary loop at a change of control mode.

### FBD/LD



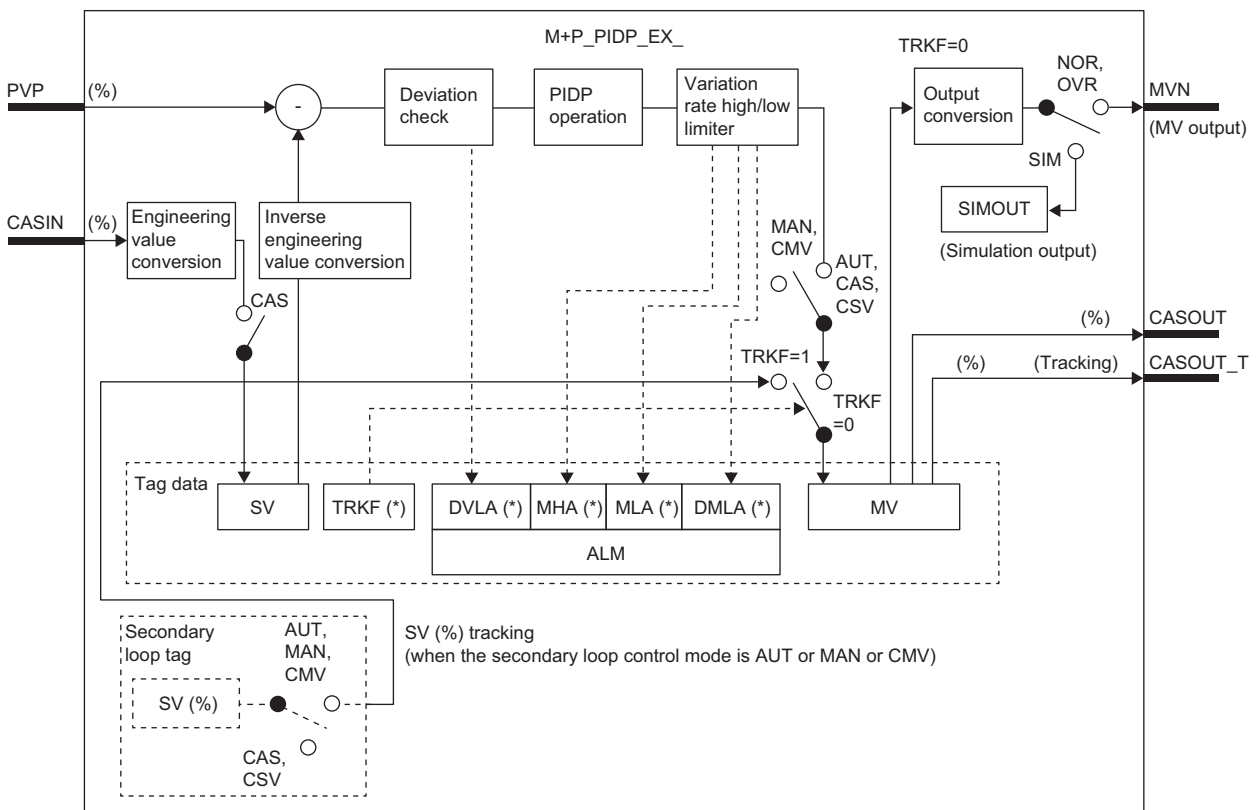
### Applicable tag type

PIDP

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bits item.

## Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN	Cascade SV input (unit: %)	0 to 100[%]	Input variable	REAL
MVN	MV output	NMIN to NMAX	Output variable	REAL
CASOUT	Cascade MV output (unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MTD	Derivative Gain	0 to 9999	8.0	User	REAL
DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
SVPTN_B0	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL

### ■Public variable (others) \*1

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

\*1 Read or write the variables using a program. They are not displayed in "FB Property" of the engineering tool.

### ■Tag data

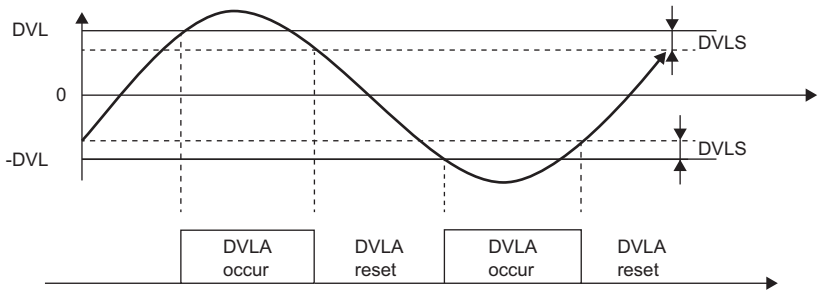
For details on tag data that is read or written with this tag access FB, refer to the following.

☞ Page 829 PIDP

## Processing details

### Deviation check

This function block performs deviation check processing.



Condition	Alarm (ALM)
	Large deviation (DVLA)
$DVL <  DV $	TRUE (Detected)
$ DV  \leq (DVL - DVLS)$	FALSE (Reset)

DV: Deviation (%), DVLS: Large deviation alarm hysteresis (%), DVL: Deviation limit value (%)

### PIDP operation

- Gain ( $K_p$ ) is calculated as follows.

$$K_p = K \times \text{PROPORTIONAL}$$

K: Output gain, PROPORTIONAL: Gain

- Output gain (K) is calculated as follows.

Condition	Output gain (K)
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is equal to 0</li> </ul>	$K = 1$
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is greater than 0</li> </ul>	<p>When <math> DV  \leq GW</math></p> $K = GG$
	<p>When <math> DV  &gt; GW</math></p> $K = 1 - \frac{(1-GG) \times GW}{ DV }$

DV: Deviation (%), GW: Gap width (%) = Rate of the gap width to the deviation, GG: Gap gain

- Deviation for PIDP operation (DV') is calculated as follows.

Condition	Deviation for PIDP operation (DV')
DV < -GW	DV' = -(GG × GW) + (DV + GW)
DV  ≤ GW	DV' = GG × DV
DV > GW	DV' = GG × GW + (DV - GW)

DV': Deviation for PIDP operation (%)  
 DV: Deviation (%)  
 GW: Gap width (%)  
 GG: Gap gain

- Deviation (DV) for direct/reverse action is calculated as follows.

Condition	Deviation (DV)
Direct action (PN = 1)	DV (%) = PVP (%) - SV (%)
Reverse action (PN = 0)	DV (%) = SV (%) - PVP (%)

DV: Deviation (%)  
 PVP (%): PV input value (%)

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit  
 RL: Engineering value low limit  
 SV: Setting value

- PIDP operations are conducted as follows.

Item	Direct action	Reverse action
Deviation (DVn)	DVn = PVn - SVn	DVn = SVn - PVn
Output variation (MV)	$MV = Kp \times \left\{ \underbrace{DV_n}_{\text{Proportional}} + \underbrace{I_n}_{\text{Integral}} + \underbrace{B_n}_{\text{Derivative}} \right\}$	
In	$I_n = I_{n-1} + \frac{CT}{Ti} \times DV_n$	
Bn	$B_n = B_{n-1} + \frac{Md \times Td}{Md \times CT + Td} \times \left\{ (PV_n - PV_{n-1}) - \frac{CT \times B_{n-1}}{Td} \right\}$	$B_n = B_{n-1} + \frac{Md \times Td}{Md \times CT + Td} \times \left\{ -(PV_n - PV_{n-1}) - \frac{CT \times B_{n-1}}{Td} \right\}$

Kp: Gain  
 Ti: Integral time  
 Td: Derivative time  
 Md: Derivative gain  
 CT: Control cycle  
 DVn: Deviation  
 DVn-1: Last deviation value  
 PVn: Process variable  
 PVn-1: Last process variable  
 SVn: Engineering value conversion processing result



The integral term and derivative term are as follows under the following conditions.

Item	Condition	Processing
Derivative term	When Td = 0	Bn = 0
	When the control mode is MAN	
	When the control mode is CMV	
Integral term	When Ti = 0	$\frac{CT}{Ti} \times DV_n = 0$
	When an MH error has occurred and the following expression is satisfied $\frac{CT}{Ti} \times DV_n > 0$	
	When an ML error has occurred and the following expression is satisfied $\frac{CT}{Ti} \times DV_n < 0$	
	When the control mode is MAN	
	When the control mode is CMV	
	When tracking from the secondary loop is performed	$I_{n-1} = \frac{MV}{Kp} - (DV_n + B_n)$
	When the control mode is changed from MAN or CMV to AUT, CAS, or CSV	

- Ti: Integral time
- CT: Control cycle
- DVn: Deviation
- MH: Output high limit value
- ML: Output low limit value

Set an integral multiple of the execution cycle ( $\Delta T$ ) as a control cycle (CT).  
 Set 0.0 or a value equal to or larger than the control cycle (CT) as an integral constant.  
 PIDP operations of this tag access FB are performed every control cycle (CT) (MV output).  
 In other execution cycles ( $\Delta T$ ), the last manipulated value is held.

### Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH-RL}{100} \times \text{Setting value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

### Inverse engineering value conversion

This function block converts the setting value (SV) of an engineering value into a setting value (SV) in percentage (%).

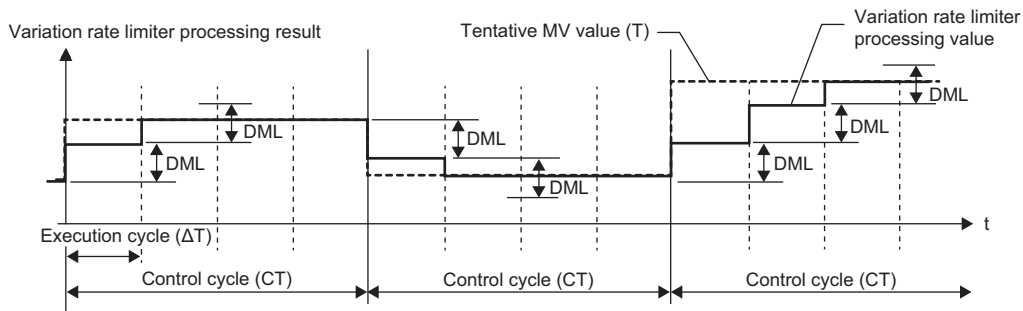
$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## Variation rate & high/low limiter

This function block checks the variation rate and high/low limits of the input value.

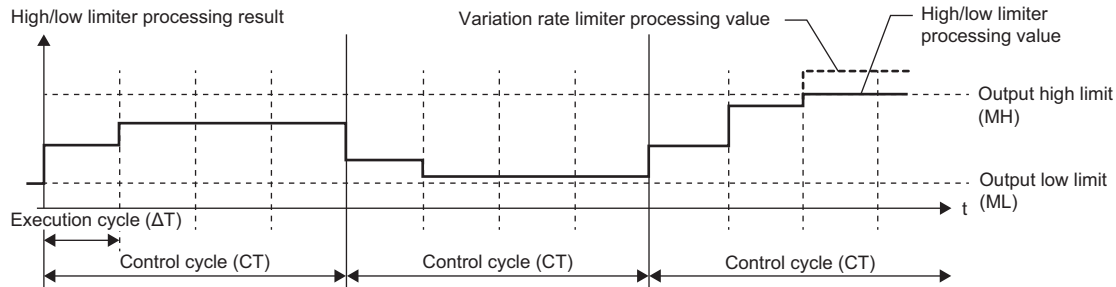
### Variation rate limiter



Condition	Variation rate limiter processing result	Alarm (ALM)
		Output variation rate limit (DMLA)
$ T - MV  \leq DML$	T	FALSE (Reset)
$T - MV > DML$	$MV + DML$	TRUE (Detected)
$T - MV < -DML$	$MV - DML$	TRUE (Detected)

T: Tentative manipulated value, MV: Manipulated value, DML: Output variation rate high limit value

### High/low limiter

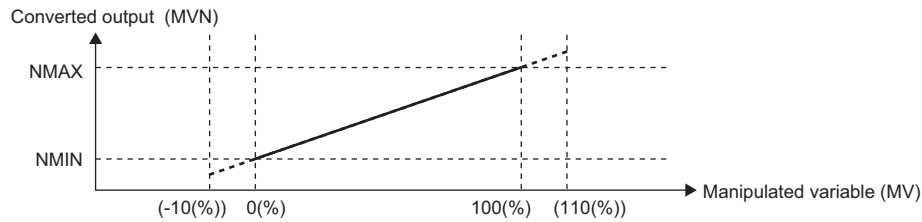


Condition	High/low limiter processing result	Alarm (ALM)	
		Output Low Limit Alarm (MLA)	Output High Limit Alarm (MHA)
Variation rate limiter processing result > MH	MH	FALSE (Reset)	TRUE (Detected)
Variation rate limiter processing result < ML	ML	TRUE (Detected)	FALSE (Reset)
$ML \leq \text{Variation rate limiter processing result} \leq MH$	Variation rate limiter processing value	FALSE (Reset)	FALSE (Reset)

MH: Output high limit value, ML: Output low limit value

## Output conversion

This function block performs output conversion processing.



$$\text{Converted output (MVN)} = \left\{ (\text{NMAX} - \text{NMIN}) \times \frac{\text{MV}}{100} \right\} + \text{NMIN}$$

NMAX: Output conversion high limit value

NMIN: Output conversion low limit

MV: Manipulated value (%)

MVN: Output conversion output value

## Disable alarm detection

This function sets whether to detect alarms (ALM) or not in the deviation check and variation rate & high/low limiter processing.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, DVLA, DMLA, MHA, or MLA of the alarm (ALM) will not be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DVLI</li> <li>• DMLI</li> <li>• MHI</li> <li>• MLI</li> </ul>
Disable alarm detection by control mode selection	In the MAN or CMV mode, DVLA, DMLA, MHA, and MLA of the alarm (ALM) are reset and the DVLA, DMLA, MHA, or MLA will not be detected.
Disable alarm detection by loop stop processing	Page 297 Loop stop processing

## Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- Outputs (MVN) are held.
- The control mode is automatically switched to MANUAL.
- When DVLA, DMLA, MHA, or MLA of the alarm (ALM) has occurred, the DVLA, DMLA, MHA, or MLA is reset.
- No alarm is detected in the variation rate & high/low limiter processing.

## Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation						
	Deviation check	PIDP operation	Engineering value conversion	Inverse engineering value conversion	Variation rate & high/low limiter	Output conversion	Alarm
MAN, CMV	○	○	×	○	×	○	○ <sup>*1</sup>
AUT	○	○	×	○	○	○	○ <sup>*2</sup>
CAS, CSV	○	○	○	○	○	○	○ <sup>*2</sup>

\*1 An alarm (ALM) whose corresponding bit is TRUE (detected) is reset, and the alarm will not be detected.

\*2 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

## Operation error

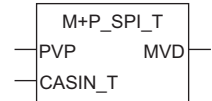
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP), operation constant (MTD, DVLS, PN, SVPTN_B0, NMAX, NMIN), or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.

# 15.13 Sample PI Control (Enable Tracking for primary loop) (M+P\_SPI\_T)

## M+P\_SPI\_T

This FB performs PI control during operating time (ST\_) and outputs the result ( $\Delta MV$ ). It holds the output ( $\Delta MV = 0$ ) during hold time (HT). The primary loop can be tracked.

### FBD/LD



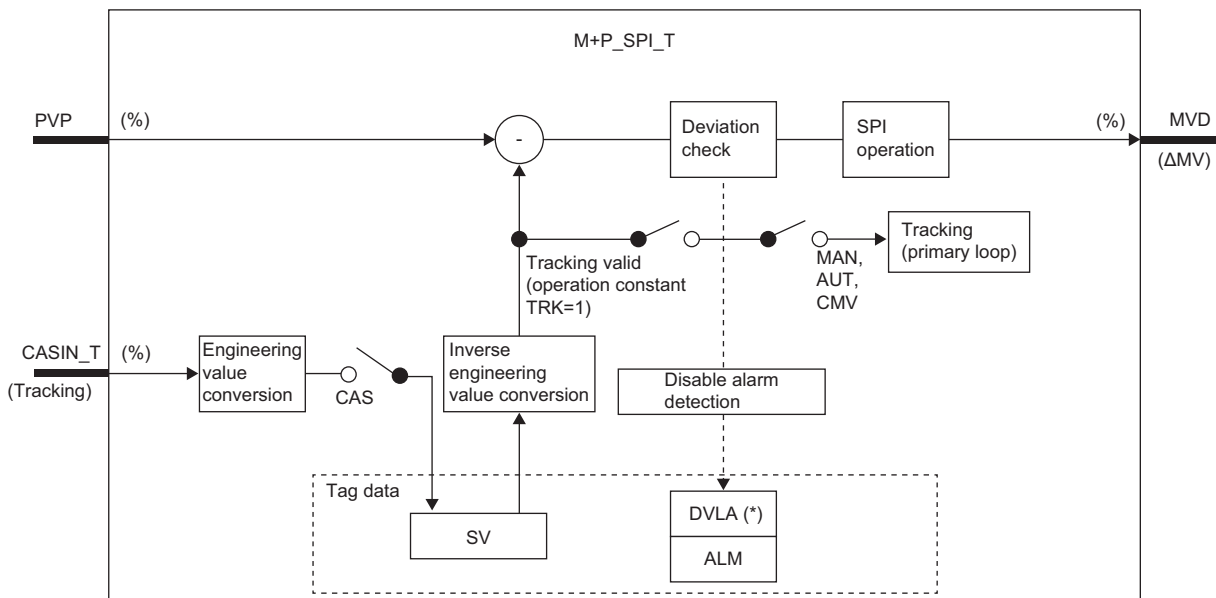
### Applicable tag type

SPI

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bits item.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN_T	Cascade SV input (unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVD	$\Delta MV$ output (unit: %)	-999999 to 999999[%]	Output variable	REAL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
SVPTN_B0	Setting value (SV) used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SVPTN_B1	Setting value (SV) pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

## Tag data

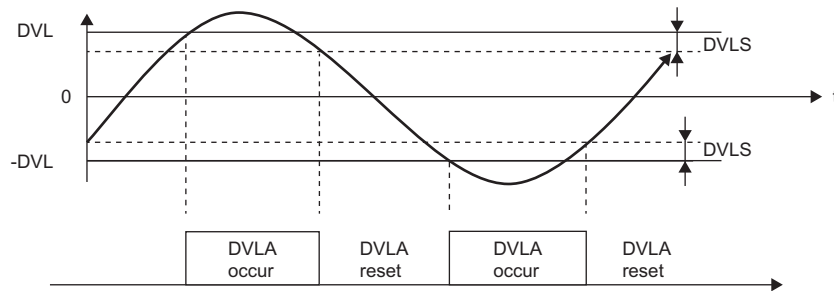
For details on tag data that is read or written with this tag access FB, refer to the following.

Page 833 SPI

## Processing details

### Deviation check

This function block performs deviation check processing.



Condition	Alarm (ALM)
	Large deviation (DVLA)
$DVL <  DV $	TRUE (Detected)
$ DV  \leq (DVL - DVLS)$	FALSE (Reset)

DV: Deviation (%), DVLS: Large deviation alarm hysteresis (%), DVL: Deviation limit value (%)

## ■SPI operation

- Gain ( $K_p$ ) is calculated as follows.

$$K_p = K \times \text{PROPORTIONAL}$$

K: Output gain, PROPORTIONAL: Gain

- Output gain (K) is calculated as follows.

Condition	Output gain (K)
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is equal to 0</li> </ul>	$K = 1$
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is greater than 0</li> </ul>	<p>When <math> DV  \leq GW</math></p> $K = GG$
	<p>When <math> DV  &gt; GW</math></p> $K = 1 - \frac{(1-GG) \times GW}{ DV }$

DV: Deviation (%), GW: Gap width (%) = Rate of the gap width to the deviation, GG: Gap gain

- Deviation for SPI operation ( $DV'$ ) is calculated as follows.

Condition	Deviation for SPI operation ( $DV'$ )
$DV < -GW$	$DV' = -(GG \times GW) + (DV + GW)$
$ DV  \leq GW$	$DV' = GG \times DV$
$DV > GW$	$DV' = GG \times GW + (DV - GW)$

$DV'$ : Deviation for SPI operation (%)

DV: Deviation (%)

GW: Gap width (%)

GG: Gap gain

- Deviation (DV) for direct/reverse action is calculated as follows.

Condition	Deviation (DV)
Direct action (PN = 1)	$DV (\%) = PVP (\%) - SV (\%)$
Reverse action (PN = 0)	$DV (\%) = SV (\%) - PVP (\%)$

DV: Deviation (%)

PVP (%): PV input value (%)

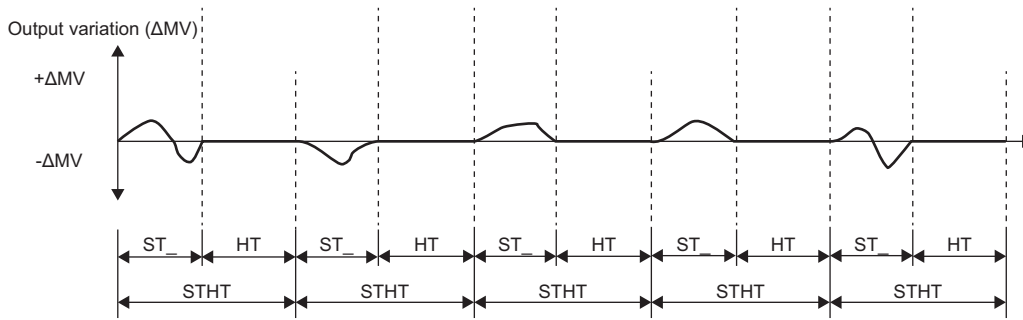
$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit

RL: Engineering value low limit

SV: Setting value

- SPI operations are conducted as follows.



ST\_: Operating time, STHT: Sample time, HT: Hold time (=STHT-ST\_)

Item	Direct action	Reverse action
Deviation (DVn)	$DVn = PVn - SVn$	$DVn = SVn - PVn$
Output variation $\Delta MV$ during operating time (ST_)	$\Delta MV = Kp \times \left\{ (DVn - DV_{n-1}) + \frac{\Delta T}{Ti} \times DVn \right\}$ <p style="text-align: center;"> <span style="margin-right: 40px;"><math>\underbrace{\hspace{1.5cm}}</math> Gain</span> <span style="margin-right: 40px;"><math>\underbrace{\hspace{1.5cm}}</math> Proportional</span> <span><math>\underbrace{\hspace{1.5cm}}</math> Integral</span> </p>	
Output variation $\Delta MV$ during hold time (HT = STHT-ST_)	$\Delta MV = 0$	

Kp: Gain  
 Ti: Integral time  
 DVn: Deviation  
 DVn-1: Last deviation value  
 PVn: Process variable  
 SVn: Engineering value conversion processing result  
 $\Delta T$ : Execution cycle  
 ST\_: Operating time  
 STHT: Sample time  
 HT: Hold time (STHT-ST\_)

The integral term is as follows under the following conditions.

Condition	Processing
When $Ti = 0$	$\frac{\Delta T}{Ti} \times DVn = 0$
When either of MH or ML error has occurred, $MVP > MH$ and the following expression is satisfied	
$\frac{\Delta T}{Ti} \times DVn > 0$	
When either of MH or ML error has occurred, $MVP < ML$ and the following expression is satisfied	$\frac{\Delta T}{Ti} \times DVn < 0$
$\frac{\Delta T}{Ti} \times DVn < 0$	

Ti: Integral time  
 $\Delta T$ : Execution cycle  
 DVn: Deviation  
 MH: Output high limit value  
 ML: Output low limit value  
 MVP: MV Internal operation value

In the following case, the hold time (HT) is handled as 0 and the PI control is continuously performed.

$$\frac{STHT}{\Delta T} \leq \frac{ST}{\Delta T}$$



## ■Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH-RL}{100} \times \text{Setting value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Inverse engineering value conversion

This function block converts the setting value (SV) of an engineering value into a setting value (SV) in percentage (%).

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value


## ■Tracking processing

The following table shows whether tracking processing to the input variable CASIN\_T is performed or not.

Condition		Result
Tracking Flag (TRK)	Setting Value (SV) Used (SVPTN_B0)	
1	FALSE	The input variable CASIN_T is tracked.
	TRUE	The input variable CASIN_T is not tracked.
0	FALSE or TRUE	

## ■Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the deviation check.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no DVLA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DVLI</li> </ul>
Disable alarm detection by loop stop processing	 Page 303 Loop stop processing

## ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- ΔMV is cleared to 0.
- The control mode is automatically switched to MANUAL.
- When DVLA of the alarm (ALM) has occurred, the DVLA is reset.
- No alarm is detected in the deviation check.

## ■Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation					
	Deviation check	SPI operation	Engineering value conversion	Inverse engineering value conversion	Tracking	Alarm
MAN, CMV, AUT	○	○	×	○	○*1	○*2
CAS, CSV	○	○	○	○	×	○*2

\*1 Tracking is performed when the tracking flag (TRK) is 1.

\*2 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

## Operation error

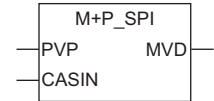
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP), operation constant (DVLS, PN, TRK, SVPTN_B0, SVPTN_B1), or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The operating time (ST_) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The sample time (STHT) setting is less than 0.
	The value divided the operating time (ST_) by the execution cycle ( $\Delta T$ ) exceeds 32767.
	The value divided the sample time (STHT) by the execution cycle ( $\Delta T$ ) exceeds 32767.

# 15.14 Sample PI Control (Disable Tracking for primary loop) (M+P\_SPI)

## M+P\_SPI

This FB performs PI control during operating time (ST\_) and outputs the result ( $\Delta MV$ ). It holds the output ( $\Delta MV = 0$ ) during hold time (HT).

### FBD/LD



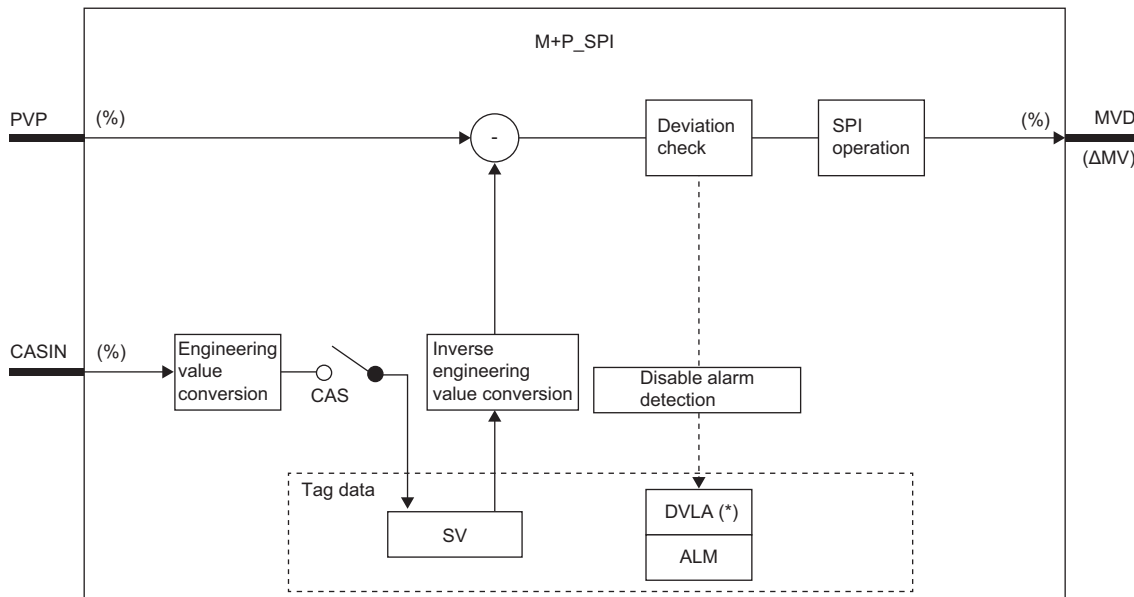
### Applicable tag type

SPI

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bits item.

### Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN	Cascade SV input (unit: %)	0 to 100[%]	Input variable	REAL
MVD	$\Delta MV$ output (unit: %)	-999999 to 999999[%]	Output variable	REAL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
SVPTN_B0	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL

## Tag data

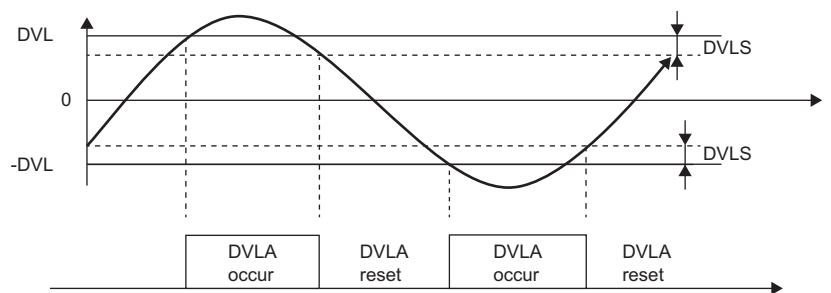
For details on tag data that is read or written with this tag access FB, refer to the following.

📄 Page 833 SPI

## Processing details

### Deviation check

This function block performs deviation check processing.



Condition	Alarm (ALM)
	Large deviation (DVLA)
$DVL <  DV $	TRUE (Detected)
$ DV  \leq (DVL - DVLS)$	FALSE (Reset)

DV: Deviation (%), DVLS: Large deviation alarm hysteresis (%), DVL: Deviation limit value (%)

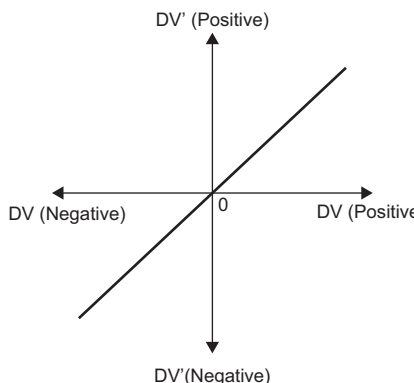
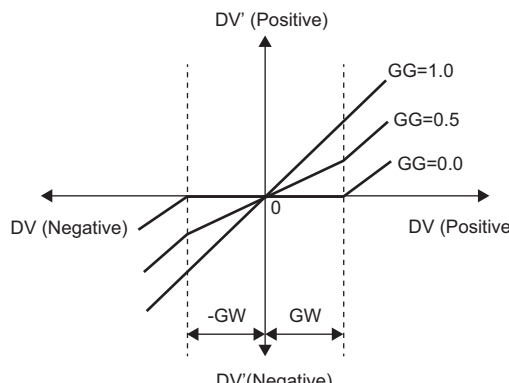
## ■SPI operation

- Gain ( $K_p$ ) is calculated as follows.

$$K_p = K \times \text{PROPORTIONAL}$$

K: Output gain, PROPORTIONAL: Gain

- Output gain (K) is calculated as follows.

Condition	Output gain (K)
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is equal to 0</li> </ul> 	$K = 1$
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is greater than 0</li> </ul> 	<p>When <math> DV  \leq GW</math></p> $K = GG$ <p>When <math> DV  &gt; GW</math></p> $K = 1 - \frac{(1-GG) \times GW}{ DV }$

DV: Deviation (%), GW: Gap width (%) = Rate of the gap width to the deviation, GG: Gap gain

- Deviation for SPI operation ( $DV'$ ) is calculated as follows.

Condition	Deviation for SPI operation ( $DV'$ )
$DV < -GW$	$DV' = -(GG \times GW) + (DV + GW)$
$ DV  \leq GW$	$DV' = GG \times DV$
$DV > GW$	$DV' = GG \times GW + (DV - GW)$

$DV'$ : Deviation for SPI operation (%)

DV: Deviation (%)

GW: Gap width (%)

GG: Gap gain

- Deviation (DV) for direct/reverse action is calculated as follows.

Condition	Deviation (DV)
Direct action (PN = 1)	$DV (\%) = PVP (\%) - SV (\%)$
Reverse action (PN = 0)	$DV (\%) = SV (\%) - PVP (\%)$

DV: Deviation (%)

PVP (%): PV input value (%)

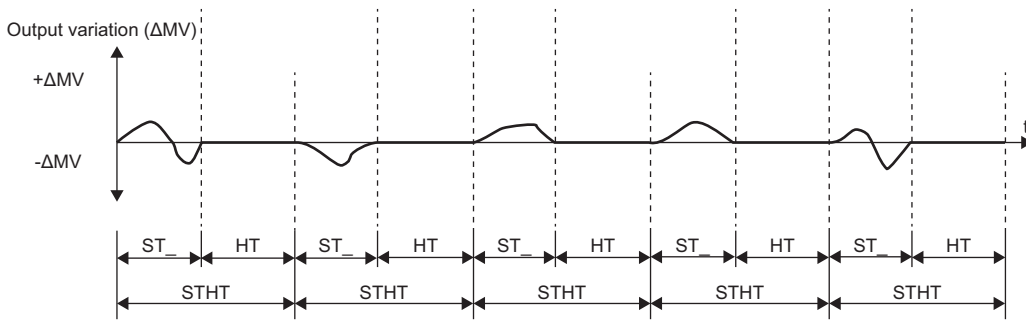
$$SV (\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit

RL: Engineering value low limit

SV: Setting value

- SPI operations are conducted as follows.



ST<sub>-</sub>: Operating time, STHT: Sample period, HT: Hold time (STHT-ST<sub>-</sub>)

Item	Direct action	Reverse action
Deviation (DV <sub>n</sub> )	DV <sub>n</sub> = PV <sub>n</sub> - SV <sub>n</sub>	DV <sub>n</sub> = SV <sub>n</sub> - PV <sub>n</sub>
Output variation ΔMV during operating time (ST <sub>-</sub> )	$\Delta MV = K_p \times \left\{ (DV_n - DV_{n-1}) + \frac{\Delta T}{T_i} \times DV_n \right\}$ <p style="text-align: center;"> <span style="margin-right: 100px;">Gain</span> <span style="margin-right: 100px;">Proportional</span> <span>Integral</span> </p>	
Output variation ΔMV during hold time (HT = STHT - ST <sub>-</sub> )	ΔMV = 0	

K<sub>p</sub>: Gain  
 T<sub>i</sub>: Integral time  
 DV<sub>n</sub>: Deviation  
 DV<sub>n-1</sub>: Last deviation value  
 PV<sub>n</sub>: Process variable  
 SV<sub>n</sub>: Engineering value conversion processing result  
 ΔT: Execution cycle  
 ST<sub>-</sub>: Operating time  
 STHT: Sample period  
 HT: Hold time (STHT - ST<sub>-</sub>)

The integral term is as follows under the following conditions.

Condition	Processing
When T <sub>i</sub> = 0	$\frac{\Delta T}{T_i} \times DV_n = 0$
When either of MH or ML error has occurred, MVP > MH and the following expression is satisfied	
$\frac{\Delta T}{T_i} \times DV_n > 0$	
When either of MH or ML error has occurred, MVP < ML and the following expression is satisfied	$\frac{\Delta T}{T_i} \times DV_n = 0$
$\frac{\Delta T}{T_i} \times DV_n < 0$	

T<sub>i</sub>: Integral time  
 ΔT: Execution cycle  
 DV<sub>n</sub>: Deviation  
 MH: Output high limit value  
 ML: Output low limit value  
 MVP: MV Internal operation value

In the following case, the hold time (HT) is handled as 0 and the PI control is continuously performed.

$$\frac{STHT}{\Delta T} \leq \frac{ST}{\Delta T}$$

### ■ Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH-RL}{100} \times \text{Setting value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

### ■Inverse engineering value conversion

This function block converts the setting value (SV) of an engineering value into a setting value (SV) in percentage (%).

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

### ■Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the deviation check.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no DVLA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DVLI</li> </ul>
Disable alarm detection by loop stop processing	Page 309 Loop stop processing

### ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- ΔMV is cleared to 0.
- The control mode is automatically switched to MANUAL.
- When DVLA of the alarm (ALM) has occurred, the DVLA is reset.
- No alarm is detected in the deviation check.

### ■Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation				
	Deviation check	SPI operation	Engineering value conversion	Inverse engineering value conversion	Alarm
MAN, CMV, AUT	○	○	×	○	○*1
CAS, CSV	○	○	○	○	○*1

\*1 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

### Operation error

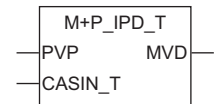
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP), operation constant (DVLS, PN, SVPTN_B0), or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The operating time (ST_) setting is less than 0.
	The execution cycle (ΔT) setting is less than 0.
	The sample time (STHT) setting is less than 0.
	The value divided the operating time (ST_) by the execution cycle (ΔT) exceeds 32767.
	The value divided the sample time (STHT) by the execution cycle (ΔT) exceeds 32767.

# 15.15 I-PD Control (Enable Tracking for primary loop) (M+P\_IPD\_T)

## M+P\_IPD\_T

This function block controls values to slowly respond to a change of the setting value without any shocks and outputs the result ( $\Delta MV$ ) by using the process variable in the proportional term and derivative term. The primary loop can be tracked.

### FBD/LD



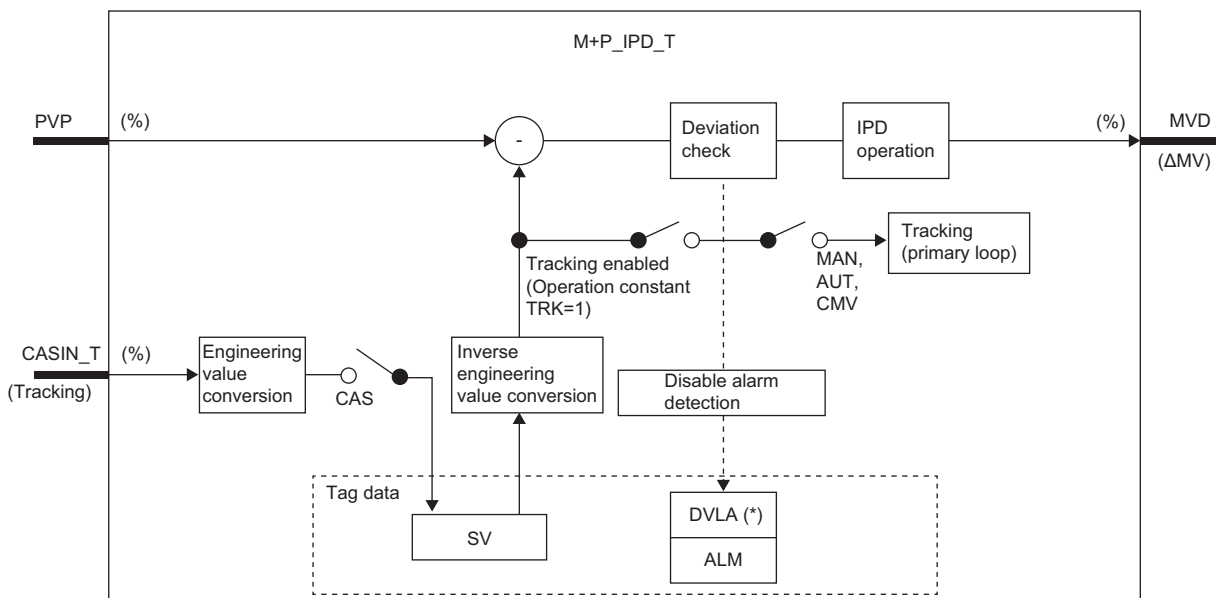
### Applicable tag type

IPD

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bits item.

### Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN_T	Cascade SV input (unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVD	$\Delta MV$ output (unit: %)	-999999 to 999999[%]	Output variable	REAL



## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MTD	Derivative Gain	0 to 9999	8.0	User	REAL
DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
SVPTN_B0	Setting value (SV) used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SVPTN_B1	Setting value (SV) pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

## Tag data

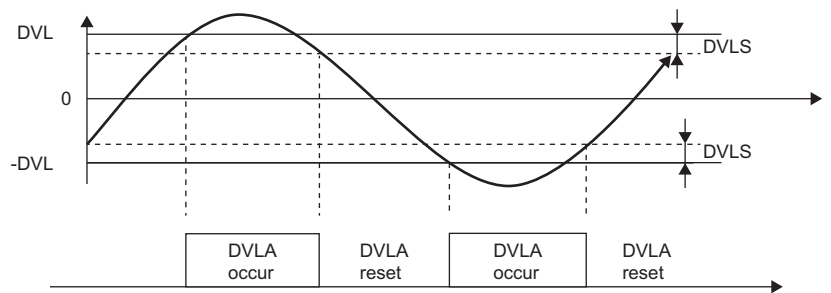
For details on tag data that is read or written with this tag access FB, refer to the following.

Page 837 IPD

## Processing details

### Deviation check

This function block performs deviation check processing.



Condition	Alarm (ALM)
	Large deviation (DVLA)
$DV <  DVL $	TRUE (Detected)
$ DV  \leq (DVL - DVLS)$	FALSE (Reset)

DV: Deviation (%), DVLS: Large deviation alarm hysteresis (%), DVL: Deviation limit value (%)

## ■IPD operation

- Gain ( $K_p$ ) is calculated as follows.

$$K_p = K \times \text{PROPORTIONAL}$$

K: Output gain, PROPORTIONAL: Gain

- Output gain (K) is calculated as follows.

Condition		Output gain (K)
<ul style="list-style-type: none"> <li>• K value to the deviation (DV) of when the gap width (GW) is equal to 0</li> </ul>		$K = 1$
<ul style="list-style-type: none"> <li>• K value to the deviation (DV) of when the gap width (GW) is greater than 0</li> </ul>	<p>When <math> DV  \leq GW</math></p>	$K = GG$
	<p>When <math> DV  &gt; GW</math></p>	$K = 1 - \frac{(1-GG) \times GW}{ DV }$

DV: Deviation (%), GW: Gap width (%) = Rate of the gap width to the deviation, GG: Gap gain

- Deviation for IPD operation ( $DV'$ ) is calculated as follows.

Condition	Deviation for IPD operation ( $DV'$ )
$DV < -GW$	$DV' = -(GG \times GW) + (DV + GW)$
$ DV  \leq GW$	$DV' = GG \times DV$
$DV > GW$	$DV' = GG \times GW + (DV - GW)$

$DV'$ : Deviation for IPD operation (%)

DV: Deviation (%)

GW: Gap width (%)

GG: Gap gain

- Deviation (DV) for direct/reverse action is calculated as follows.

Condition	Deviation (DV)
Direct action (PN = 1)	$DV (\%) = PVP (\%) - SV (\%)$
Reverse action (PN = 0)	$DV (\%) = SV (\%) - PVP (\%)$

DV: Deviation (%)

PVP (%): PV input value (%)

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit

RL: Engineering value low limit

SV: Setting value

- IPD operations are conducted as follows.

Item	Direct action	Reverse action
Deviation (DVn)	$DVn = PVn - SVn$	$DVn = SVn - PVn$
Output variation ( $\Delta MV$ )	$\Delta MV = K_p \times \left\{ \underbrace{\frac{CT}{T_i}}_{\text{Gain}} \times DV_n + \underbrace{(PV_n - PV_{n-1})}_{\text{Integral}} + \underbrace{B_n}_{\text{Proportional}} + \underbrace{\quad}_{\text{Derivative}} \right\}$	$\Delta MV = K_p \times \left\{ \underbrace{\frac{CT}{T_i}}_{\text{Gain}} \times DV_n - \underbrace{(PV_n - PV_{n-1})}_{\text{Integral}} + \underbrace{B_n}_{\text{Proportional}} + \underbrace{\quad}_{\text{Derivative}} \right\}$
Bn	$B_n = B_{n-1} + \frac{Md \times T_d}{Md \times CT + T_d} \times \left\{ (PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times B_{n-1}}{T_d} \right\}$	$B_n = B_{n-1} + \frac{Md \times T_d}{Md \times CT + T_d} \times \left\{ -(PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times B_{n-1}}{T_d} \right\}$

Kp: Gain  
 Ti: Integral time  
 Td: Derivative time  
 Md: Derivative gain  
 CT: Control cycle  
 DVn: Deviation  
 DVn-1: Last deviation value  
 PVn: Process variable  
 PVn-1: Last process variable  
 PVn-2: Process variable before the last value  
 SVn: Engineering value conversion processing result

The integral term and derivative term are as follows under the following conditions.

Item	Condition	Processing
Derivative term	When Td = 0	Bn = 0
	When the control mode is MAN	
	When the control mode is CMV	
Integral term	When Ti = 0	$\frac{CT}{T_i} \times DV_n = 0$
	When either of MH or ML error has occurred, MVP > MH and the following expression is satisfied $\frac{CT}{T_i} \times DV_n > 0$	
	• When either of MH or ML error has occurred, MVP < ML and the following expression is satisfied $\frac{CT}{T_i} \times DV_n < 0$	

Ti: Integral time  
 CT: Control cycle  
 DVn: Deviation  
 MH: Output high limit value  
 ML: Output low limit value  
 MVP: MV Internal operation value

Set an integral multiple of the execution cycle ( $\Delta T$ ) as a control cycle (CT).

Set 0.0 or a value equal to or larger than the control cycle (CT) as an integral constant.

PID operations of this tag access FB are performed every control cycle (CT) ( $\Delta MV$  output).

In other execution cycles ( $\Delta T$ ), the last value is held ( $\Delta MV = 0$ ).

## ■Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH-RL}{100} \times \text{Setting value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Inverse engineering value conversion

This function block converts the setting value (SV) of an engineering value into a setting value (SV) in percentage (%).

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value


## ■Tracking processing

The following table shows whether tracking processing to the input variable CASIN\_T is performed or not.

Condition		Result
Tracking Flag (TRK)	Setting Value (SV) Used (SVPTN_B0)	
1	FALSE	The input variable CASIN_T is tracked.
	TRUE	The input variable CASIN_T is not tracked.
0	FALSE or TRUE	

## ■Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the deviation check.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no DVLA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DVLI</li> </ul>
Disable alarm detection by loop stop processing	 Page 314 Loop stop processing

## ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- ΔMV is cleared to 0.
- The control mode is automatically switched to MANUAL.
- When DVLA of the alarm (ALM) has occurred, the DVLA is reset.
- No alarm is detected in the deviation check.

## ■Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation					
	Deviation check	IPD operation	Engineering value conversion	Inverse engineering value conversion	Tracking	Alarm
MAN, CMV, AUT	○	○	×	○	○ <sup>*1</sup>	○ <sup>*2</sup>
CAS, CSV	○	○	○	○	×	○ <sup>*2</sup>

\*1 Tracking is performed when the tracking flag (TRK) is 1.

\*2 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

## Operation error

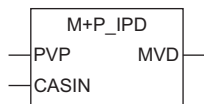
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP), operation constant (MTD, DVLS, PN, TRK, SVPTN_B0, SVPTN_B1), or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.

# 15.16 I-PD Control (Disable Tracking for primary loop) (M+P\_IPD)

## M+P\_IPD

This function block controls values to slowly respond to a change of the setting value without any shocks and outputs the result ( $\Delta MV$ ) by using the process variable in the proportional term and derivative term.

### FBD/LD



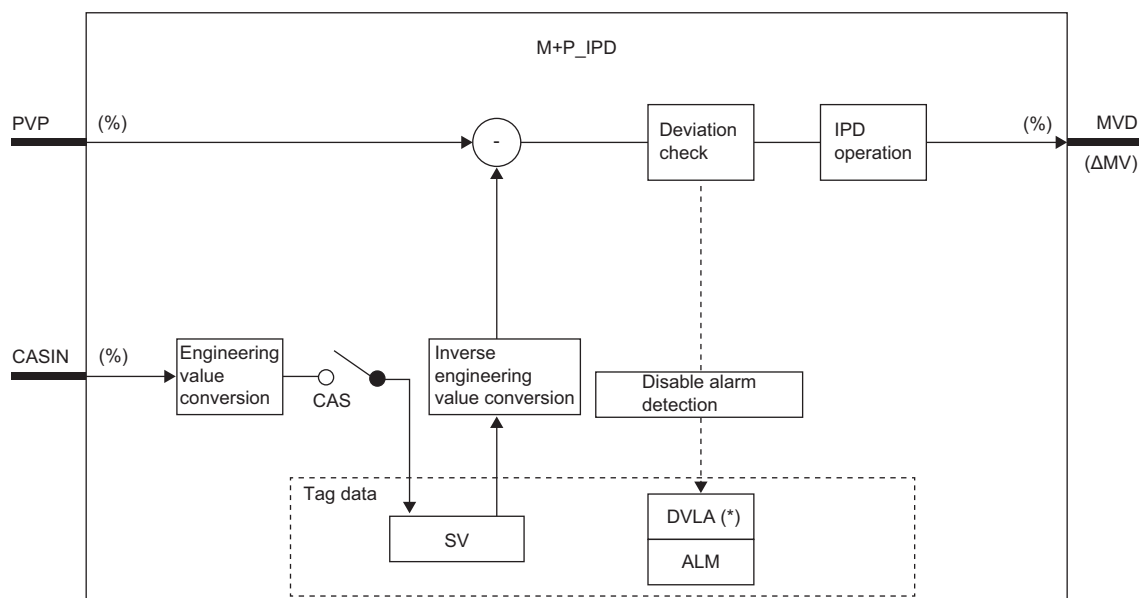
### Applicable tag type

IPD

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bits item.

### Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN	Cascade SV input (unit: %)	0 to 100[%]	Input variable	REAL
MVD	$\Delta MV$ output (unit: %)	-999999 to 999999[%]	Output variable	REAL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MTD	Derivative Gain	0 to 9999	8.0	User	REAL
DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
SVPTN_B0	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL

## Tag data

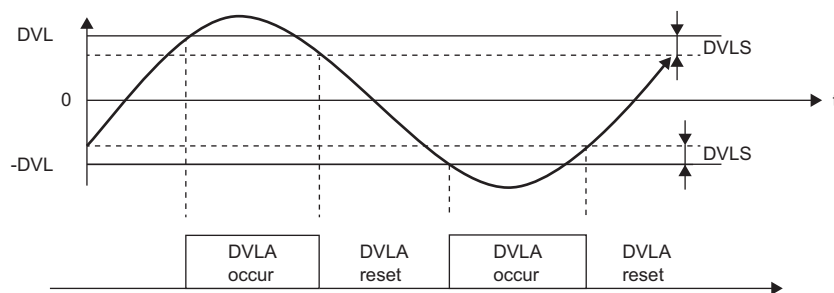
For details on tag data that is read or written with this tag access FB, refer to the following.

Page 837 IPD

## Processing details

### Deviation check

This function block performs deviation check processing.



Condition	Alarm (ALM)
	Large deviation (DVLA)
$DVL <  DV $	TRUE (Detected)
$ DV  \leq (DVL - DVLS)$	FALSE (Reset)

DV: Deviation (%), DVLS: Large deviation alarm hysteresis (%), DVL: Deviation limit value (%)

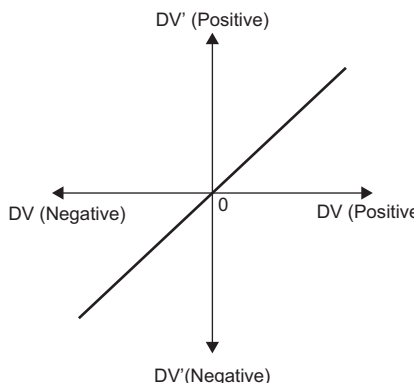
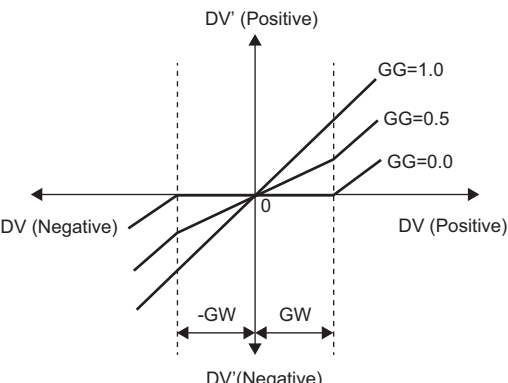
## ■IPD operation

- Gain ( $K_p$ ) is calculated as follows.

$$K_p = K \times \text{PROPORTIONAL}$$

K: Output gain, PROPORTIONAL: Gain

- Output gain (K) is calculated as follows.

Condition		Output gain (K)
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is equal to 0</li> </ul> 		$K = 1$
<ul style="list-style-type: none"> <li>K value to the deviation (DV) of when the gap width (GW) is greater than 0</li> </ul> 	When $ DV  \leq GW$ When $ DV  > GW$	$K = GG$ $K = 1 - \frac{(1-GG) \times GW}{ DV }$

DV: Deviation (%), GW: Gap width (%) = Rate of the gap width to the deviation, GG: Gap gain

- Deviation for IPD operation ( $DV'$ ) is calculated as follows.

Condition	Deviation for IPD operation ( $DV'$ )
$DV < -GW$	$DV' = -(GG \times GW) + (DV + GW)$
$ DV  \leq GW$	$DV' = GG \times DV$
$DV > GW$	$DV' = GG \times GW + (DV - GW)$

$DV'$ : Deviation for IPD operation (%)

DV: Deviation (%)

GW: Gap width (%)

GG: Gap gain

- Deviation (DV) for direct/reverse action is calculated as follows.

Condition	Deviation (DV)
Direct action (PN = 1)	$DV (\%) = PVP (\%) - SV (\%)$
Reverse action (PN = 0)	$DV (\%) = SV (\%) - PVP (\%)$

DV: Deviation (%)

PVP (%): PV input value (%)

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit

RL: Engineering value low limit

SV: Setting value



- IPD operations are conducted as follows.

Item	Direct action	Reverse action
Deviation (DVn)	$DVn = PVn - SVn$	$DVn = SVn - PVn$
Output variation ( $\Delta MV$ )	$\Delta MV = K_p \times \left\{ \frac{CT}{T_i} \times DV_n + \frac{(PV_n - PV_{n-1})}{CT} + B_n \right\}$ <p style="text-align: center;"> <span style="margin-right: 40px;">Gain</span> <span style="margin-right: 40px;">Integral</span> <span style="margin-right: 40px;">Proportional</span> <span>Derivative</span> </p>	$\Delta MV = K_p \times \left\{ \frac{CT}{T_i} \times DV_n - \frac{(PV_n - PV_{n-1})}{CT} + B_n \right\}$ <p style="text-align: center;"> <span style="margin-right: 40px;">Gain</span> <span style="margin-right: 40px;">Integral</span> <span style="margin-right: 40px;">Proportional</span> <span>Derivative</span> </p>
Bn	$B_n = B_{n-1} + \frac{Md \times T_d}{Md \times CT + T_d} \times \left\{ (PV_{n-2} - PV_{n-1} + PV_{n-2}) - \frac{CT \times B_{n-1}}{T_d} \right\}$	$B_n = B_{n-1} + \frac{Md \times T_d}{Md \times CT + T_d} \times \left\{ -(PV_{n-2} - PV_{n-1} + PV_{n-2}) - \frac{CT \times B_{n-1}}{T_d} \right\}$

Kp: Gain  
 Ti: Integral time  
 Td: Derivative time  
 Md: Derivative gain  
 CT: Control cycle  
 DVn: Deviation  
 DVn-1: Last deviation value  
 PVn: Process variable  
 PVn-1: Last process variable  
 PVn-2: Process variable before the last value  
 SVn: Engineering value conversion processing result

The integral term and derivative term are as follows under the following conditions.

Item	Condition	Processing
Derivative term	When $T_d = 0$	$B_n = 0$
	When the control mode is MAN	
	When the control mode is CMV	
Integral term	When $T_i = 0$	$\frac{CT}{T_i} \times DV_n = 0$
	When either of MH or ML error has occurred, $MVP > MH$ and the following expression is satisfied	
	When either of MH or ML error has occurred, $MVP < ML$ and the following expression is satisfied	

Ti: Integral time  
 CT: Control cycle  
 DVn: Deviation  
 MH: Output high limit value  
 ML: Output low limit value  
 MVP: MV Internal operation value

Set an integral multiple of the execution cycle ( $\Delta T$ ) as a control cycle (CT).

Set 0.0 or a value equal to or larger than the control cycle (CT) as an integral constant.

PID operations of this tag access FB are performed every control cycle (CT) ( $\Delta MV$  output).

In other execution cycles ( $\Delta T$ ), the last value is held ( $\Delta MV = 0$ ).

## ■Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH-RL}{100} \times \text{Setting value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Inverse engineering value conversion


This function block converts the setting value (SV) of an engineering value into a setting value (SV) in percentage (%).

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the deviation check.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no DVLA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DVLI</li> </ul>
Disable alarm detection by loop stop processing	 Page 320 Loop stop processing

## ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- ΔMV is cleared to 0.
- The control mode is automatically switched to MANUAL.
- When DVLA of the alarm (ALM) has occurred, the DVLA is reset.
- No alarm is detected in the deviation check.

## ■Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation				
	Deviation check	IPD operation	Engineering value conversion	Inverse engineering value conversion	Alarm
MAN, CMV, AUT	○	○	×	○	○*1
CAS, CSV	○	○	○	○	○*1

\*1 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

## Operation error

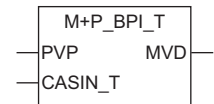
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP), operation constant (MTD, DVLS, PN, SVPTN_B0), or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle (ΔT) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle (ΔT) exceeds 32767.

# 15.17 Blend PI Control (Enable Tracking for primary loop) (M+P\_BPI\_T)

## M+P\_BPI\_T

This FB is used when the control amount vibrates in a short period but is stable in a long period. The primary loop can be tracked.

### FBD/LD



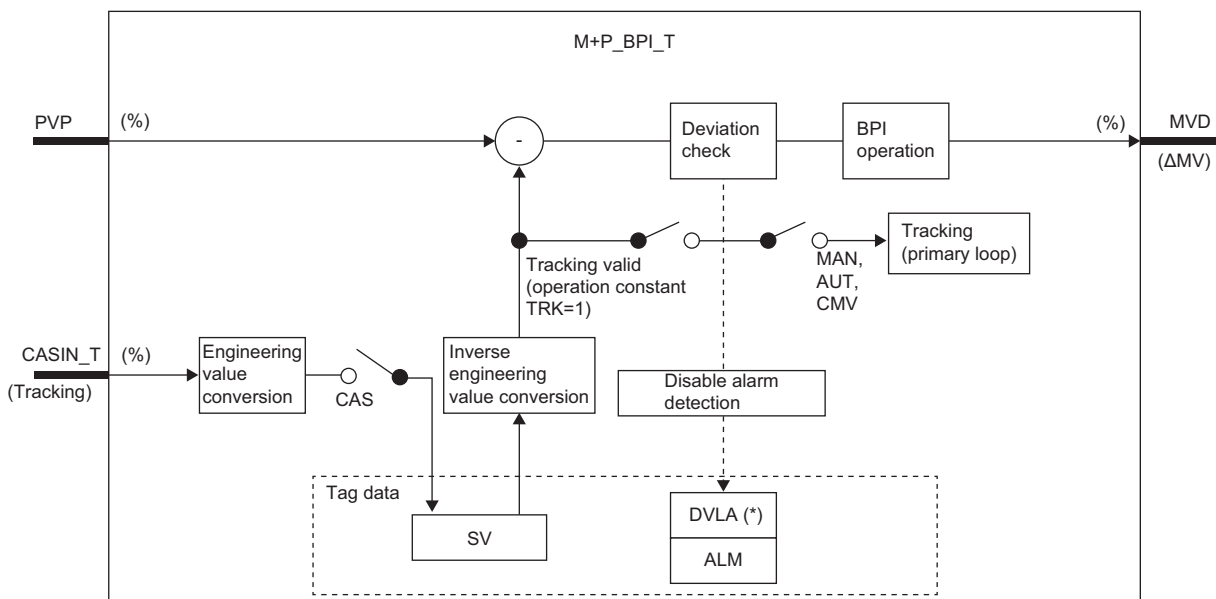
### Applicable tag type

BPI

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bits item.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN_T	Cascade SV input (unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVD	ΔMV output (unit: %)	-999999 to 999999[%]	Output variable	REAL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
SVPTN_B0	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SVPTN_B1	Setting Value (SV) Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
RST_SDV_ON_CHGMODE	Reset DV Cumulative Value in Control Mode Change	TRUE: DV cumulative value (SDV) reset at control mode change (from MAN/CMV to AUT/CAS/CSV) FALSE: DV cumulative value (SDV) not reset	FALSE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

## Public variable (others) <sup>\*1</sup>

Variable name	Description	Recommended range	Initial value	Set by	Data type
RST_SDV	Reset DV Cumulative Value FALSE → TRUE: DV cumulative value (SDV) reset	TRUE, FALSE	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed in "FB Property" of the engineering tool.

## Tag data

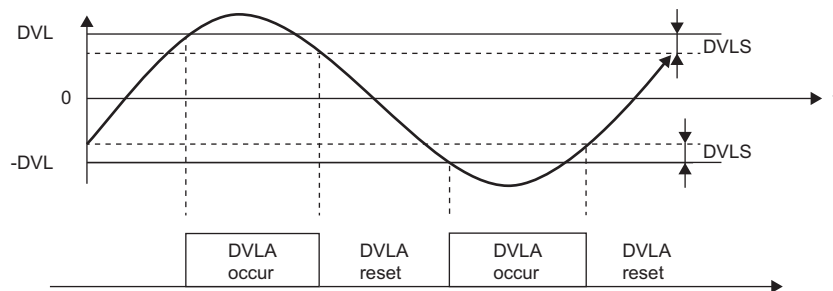
For details on tag data that is read or written with this tag access FB, refer to the following.

Page 841 BPI

## Processing details

### Deviation check

This function block performs deviation check processing.



Condition	Alarm (ALM)
	Large deviation (DVLA)
$DVL <  DV $	TRUE (Detected)
$ DV  \leq (DVL - DVLS)$	FALSE (Reset)

DV: Deviation (%), DVLS: Large deviation alarm hysteresis (%), DVL: Deviation limit value (%)

## ■BPI operation

- Gain ( $K_p$ ) is calculated as follows.

$$K_p = K \times \text{PROPORTIONAL}$$

K: Output gain, PROPORTIONAL: Gain

- Output gain (K) is calculated as follows.

Condition		Output gain (K)
<ul style="list-style-type: none"> <li>• K value to the deviation (DV) of when the gap width (GW) is equal to 0</li> </ul>		$K = 1$
<ul style="list-style-type: none"> <li>• K value to the deviation (DV) of when the gap width (GW) is greater than 0</li> </ul>	When $ DV  \leq GW$ When $ DV  > GW$	$K = GG$ $K = 1 - \frac{(1-GG) \times GW}{ DV }$

DV: Deviation (%), GW: Gap width (%) = Rate of the gap width to the deviation, GG: Gap gain

- Deviation for BPI operation ( $DV'$ ) is calculated as follows.

Condition	Deviation for BPI operation ( $DV'$ )
$DV < -GW$	$DV' = -(GG \times GW) + (DV + GW)$
$ DV  \leq GW$	$DV' = GG \times DV$
$DV > GW$	$DV' = GG \times GW + (DV - GW)$

$DV'$ : Deviation for BPI operation (%)

DV: Deviation (%)

GW: Gap width (%)

GG: Gap gain

- Deviation (DV) for direct/reverse action is calculated as follows.

Condition	Deviation (DV)
Direct action (PN = 1)	$DV (\%) = PVP (\%) - SV (\%)$
Reverse action (PN = 0)	$DV (\%) = SV (\%) - PVP (\%)$

DV: Deviation (%)

PVP (%): PV input value (%)

$$SV (\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit

RL: Engineering value low limit

SV: Setting value

- BPI operations are conducted as follows.

Item	Direct action	Reverse action
Deviation (DV <sub>n</sub> )	DV <sub>n</sub> = PV <sub>n</sub> - SV <sub>n</sub>	DV <sub>n</sub> = SV <sub>n</sub> - PV <sub>n</sub>
Output variation (ΔMV)	$\Delta MV = K_p \times \Delta T \times \left\{ DV_n + \frac{CT}{T_i} \times \Sigma DV_i \right\}$	

K<sub>p</sub>: Gain  
 T<sub>i</sub>: Integral time  
 ΔT: Execution cycle  
 CT: Control cycle  
 ΣDV<sub>i</sub>: DV<sub>n</sub> cumulative value  
 DV<sub>n</sub>: Deviation  
 PV<sub>n</sub>: Process variable  
 SV<sub>n</sub>: Engineering value conversion processing result

The integral term and derivative term are as follows under the following conditions.

Condition	Processing
T <sub>i</sub> = 0	$\frac{CT}{T_i} \times \Sigma DV_i =$ Hold the previous value
When MLA is TRUE	
When MHA is TRUE	
T <sub>i</sub> ≠ 0	$\frac{CT}{T_i} \times \Sigma DV_i = \frac{CT}{T_i} \times (\Sigma DV_i + DV_n)$

T<sub>i</sub>: Integral time  
 CT: Control cycle  
 ΣDV<sub>i</sub>: DV<sub>n</sub> cumulative value  
 DV<sub>n</sub>: Deviation  
 MLA: Output low limit alarm  
 MHA: Output high limit alarm

Set an integral multiple of the execution cycle (ΔT) as a control cycle (CT).

Set 0.0 or a value equal to or larger than the control cycle (CT) as an integral constant.

PID operations of this tag access FB are performed every control cycle (CT) (ΔMV output).

In other execution cycles (ΔT), the last value is held (ΔMV = 0).

### ■Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH-RL}{100} \times \text{Setting value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

### ■Inverse engineering value conversion

This function block converts the setting value (SV) of an engineering value into a setting value (SV) in percentage (%).

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value


### ■Tracking processing

The following table shows whether tracking processing to the input variable CASIN\_T is performed or not.

Condition		Result
Tracking Flag (TRK)	Setting Value (SV) Used (SVPTN_B0)	
1	FALSE	The input variable CASIN_T is tracked.
	TRUE	The input variable CASIN_T is not tracked.
0	FALSE or TRUE	

## Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the deviation check.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no DVLA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DVLI</li> </ul>
Disable alarm detection by loop stop processing	 Page 325 Loop stop processing

## Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- $\Delta MV$  is cleared to 0.
- The control mode is automatically switched to MANUAL.
- When DVLA of the alarm (ALM) has occurred, the DVLA is reset.
- No alarm is detected in the deviation check.

## Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation					
	Deviation check	BPI operation	Engineering value conversion	Inverse engineering value conversion	Tracking	Alarm
MAN, CMV, AUT	○	○	×	○	○ <sup>*1</sup>	○ <sup>*2</sup>
CAS, CSV	○	○	○	○	×	○ <sup>*2</sup>

\*1 Tracking is performed when the tracking flag (TRK) is 1.

\*2 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

## Operation error

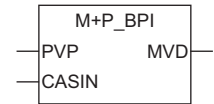
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP), operation constant (MTD, DVLS, PN, TRK, SVPTN_B0, SVPTN_B1), or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.

# 15.18 Blend PI Control (Disable Tracking for primary loop) (M+P\_BPI)

## M+P\_BPI

This FB is used when the control amount vibrates in a short period but is stable in a long period.

### FBD/LD



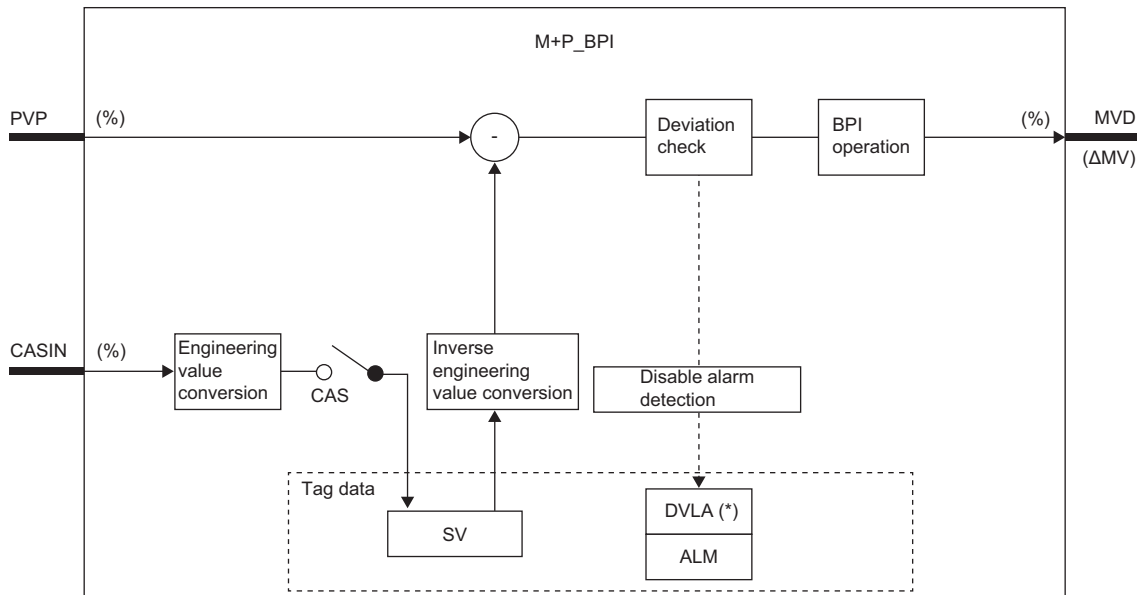
### Applicable tag type

BPI

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bits item.

### Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN	Cascade SV input (unit: %)	0 to 100[%]	Input variable	REAL
MVD	ΔMV output (unit: %)	-999999 to 999999[%]	Output variable	REAL



## Public variable (operation constant)

- Operation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
SVPTN_B0	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
RST_SDV_ON_CHGMODE	Reset DV Cumulative Value in Control Mode Change	TRUE: DV cumulative value (SDV) reset at control mode change (from MAN/CMV to AUT/CAS/CSV) FALSE: DV cumulative value (SDV) not reset	FALSE	User	BOOL

## Public variable (others) \*1

Variable name	Description	Recommended range	Initial value	Set by	Data type
RST_SDV	Reset DV Cumulative Value FALSE → TRUE: DV cumulative value (SDV) reset	TRUE, FALSE	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed in "FB Property" of the engineering tool.

## Tag data

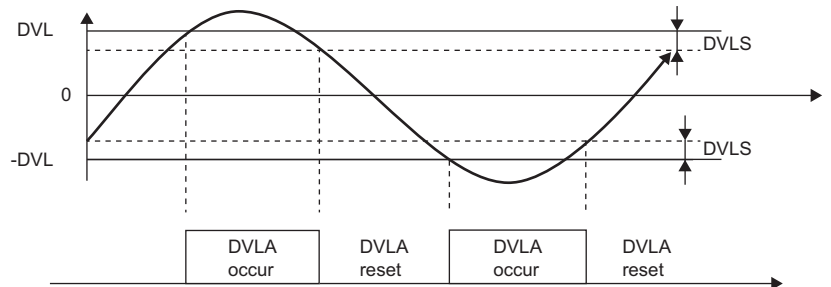
For details on tag data that is read or written with this tag access FB, refer to the following.

📄 Page 841 BPI

## Processing details

### Deviation check

This function block performs deviation check processing.



Condition	Alarm (ALM)
	Large deviation (DVLA)
$DVL <  DV $	TRUE (Detected)
$ DV  \leq (DVL - DVLS)$	FALSE (Reset)

DV: Deviation (%), DVLS: Large deviation alarm hysteresis (%), DVL: Deviation limit value (%)

## ■BPI operation

- Gain ( $K_p$ ) is calculated as follows.

$$K_p = K \times \text{PROPORTIONAL}$$

K: Output gain, PROPORTIONAL: Gain

- Output gain (K) is calculated as follows.

Condition		Output gain (K)
<ul style="list-style-type: none"> <li>• K value to the deviation (DV) of when the gap width (GW) is equal to 0</li> </ul>		$K = 1$
<ul style="list-style-type: none"> <li>• K value to the deviation (DV) of when the gap width (GW) is greater than 0</li> </ul>	When $ DV  \leq GW$ When $ DV  > GW$	$K = GG$ $K = 1 - \frac{(1-GG) \times GW}{ DV }$

DV: Deviation (%), GW: Gap width (%) = Rate of the gap width to the deviation, GG: Gap gain

- Deviation for BPI operation ( $DV'$ ) is calculated as follows.

Condition	Deviation for BPI operation ( $DV'$ )
$DV < -GW$	$DV' = -(GG \times GW) + (DV + GW)$
$ DV  \leq GW$	$DV' = GG \times DV$
$DV > GW$	$DV' = GG \times GW + (DV - GW)$

$DV'$ : Deviation for BPI operation (%)

DV: Deviation (%)

GW: Gap width (%)

GG: Gap gain

- Deviation (DV) for direct/reverse action is calculated as follows.

Condition	Deviation (DV)
Direct action (PN = 1)	$DV (\%) = PVP (\%) - SV (\%)$
Reverse action (PN = 0)	$DV (\%) = SV (\%) - PVP (\%)$

DV: Deviation (%)

PVP (%): PV input value (%)

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit

RL: Engineering value low limit

SV: Setting value

- BPI operations are conducted as follows.

Item	Direct action	Reverse action
Deviation (DV <sub>n</sub> )	DV <sub>n</sub> = PV <sub>n</sub> - SV <sub>n</sub>	DV <sub>n</sub> = SV <sub>n</sub> - PV <sub>n</sub>
Output variation (ΔMV)	$\Delta MV = K_p \times \Delta T \times \left\{ DV_n + \frac{CT}{T_i} \times \Sigma DV_i \right\}$	

K<sub>p</sub>: Gain  
 T<sub>i</sub>: Integral time  
 ΔT: Execution cycle  
 CT: Control cycle  
 ΣDV<sub>i</sub>: DV<sub>n</sub> cumulative value  
 DV<sub>n</sub>: Deviation  
 PV<sub>n</sub>: Process variable  
 SV<sub>n</sub>: Engineering value conversion processing result

The integral term and derivative term are as follows under the following conditions.

Condition	Processing
T <sub>i</sub> = 0	$\frac{CT}{T_i} \times \Sigma DV_i =$ Hold the previous value
When MLA is TRUE	
When MHA is TRUE	
T <sub>i</sub> ≠ 0	$\frac{CT}{T_i} \times \Sigma DV_i = \frac{CT}{T_i} \times (\Sigma DV_i + DV_n)$

T<sub>i</sub>: Integral time  
 CT: Control cycle  
 ΣDV<sub>i</sub>: DV<sub>n</sub> cumulative value  
 DV<sub>n</sub>: Deviation  
 MLA: Output low limit alarm  
 MHA: Output high limit alarm

Set an integral multiple of the execution cycle (ΔT) as a control cycle (CT).  
 Set 0.0 or a value equal to or larger than the control cycle (CT) as an integral constant.  
 PID operations of this tag access FB are performed every control cycle (CT) (ΔMV output).  
 In other execution cycles (ΔT), the last value is held (ΔMV = 0).

### ■Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH-RL}{100} \times \text{Setting value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

### ■Inverse engineering value conversion


This function block converts the setting value (SV) of an engineering value into a setting value (SV) in percentage (%).

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the deviation check.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no DVLA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DVLI</li> </ul>
Disable alarm detection by loop stop processing	 Page 330 Loop stop processing

## ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- $\Delta MV$  is cleared to 0.
- The control mode is automatically switched to MANUAL.
- When DVLA of the alarm (ALM) has occurred, the DVLA is reset.
- No alarm is detected in the deviation check.

## ■Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation				
	Deviation check	BPI operation	Engineering value conversion	Inverse engineering value conversion	Alarm
MAN, CMV, AUT	○	○	×	○	○*1
CAS, CSV	○	○	○	○	○*1

\*1 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

## Operation error

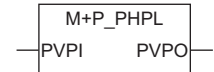
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP), operation constant (MTD, DVLS, PN, SVPTN_B0), or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.

# 15.19 High/Low Limit Alarm Check (M+P\_PHPL)

## M+P\_PHPL

This FB performs the high high limit/high limit/low limit/low low limit checks and the variation rate check to the input (PVPI) and outputs the results. If a value has exceeded an allowable range, an alarm occurs.

### FBD/LD



### Applicable tag type

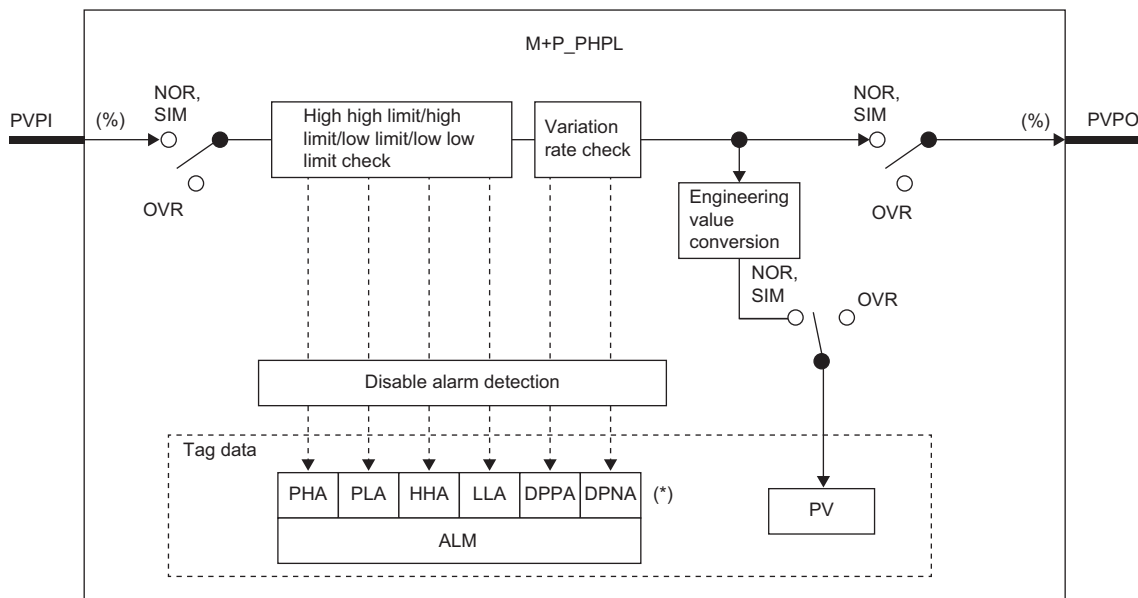
PID, 2PID, 2PIDH, PIDP, SPI, IPD, BPI, R, ONF2, ONF3, MONI, SWM, MWM, PVAL

### Control mode

MAN	AUT	CAS*1	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

\*1 Transition to CASDR is possible.

### Block diagram



\* Indicates bits item.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVPI	PV input (unit: %)	0 to 100[%]	Input variable	REAL
PVPO	PV output (unit: %)	0 to 100[%]	Output variable	REAL

#### Tag data

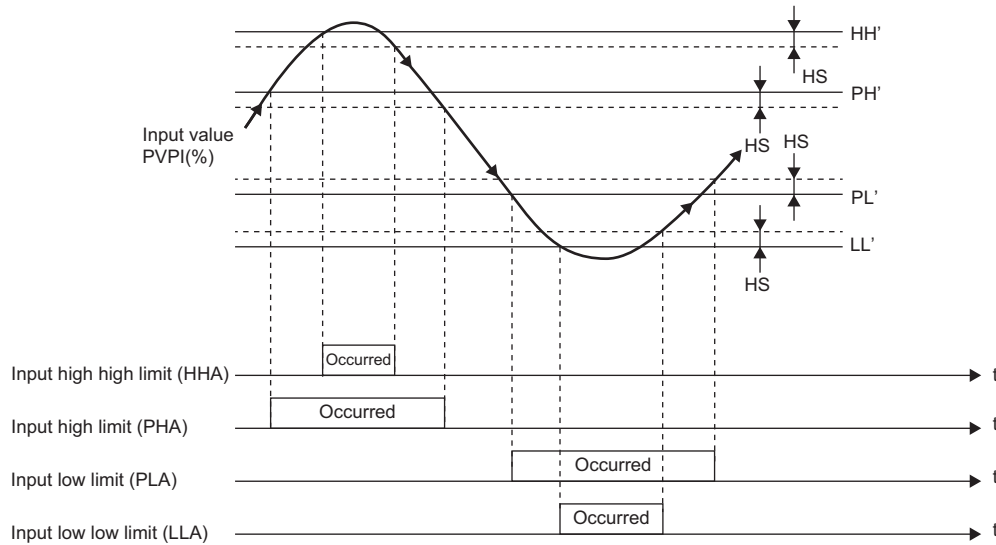
For details on tag data that is read or written with this tag access FB, refer to the following.

☞ Page 812 Tag Data List

## Processing details

### High high limit/high limit/low limit/low low limit check

This function block performs the high/low limit checks to the input value.



Item		Input condition of when an alarm occurs	
		TRUE (Detected)	FALSE (Reset)
Alarm (ALM)	Input high high limit (HHA)	Input value (%) > HH'	Input value (%) ≤ HH' - HS
	Input high limit (PHA)	Input value (%) > PH'	Input value (%) ≤ PH' - HS
	Input low limit (PLA)	Input value (%) < PL'	Input value (%) ≥ PL' + HS
	Input low low limit (LLA)	Input value (%) < LL'	Input value (%) ≥ LL' + HS

HH: High high limit alarm value

PH\_: High limit alarm value

PL: Low limit alarm value

LL: Low low limit alarm value

HH': High high limit alarm value (%)

PH': High limit alarm value (%)

PL': Low limit alarm value (%)

LL': Low low limit alarm value (%)

HS: High/low limit alarm hysteresis (%)

- The high high limit/high limit/low limit/low low limit alarm values are converted into values in percentage (%) in accordance with the input (Inverse engineering value conversion)

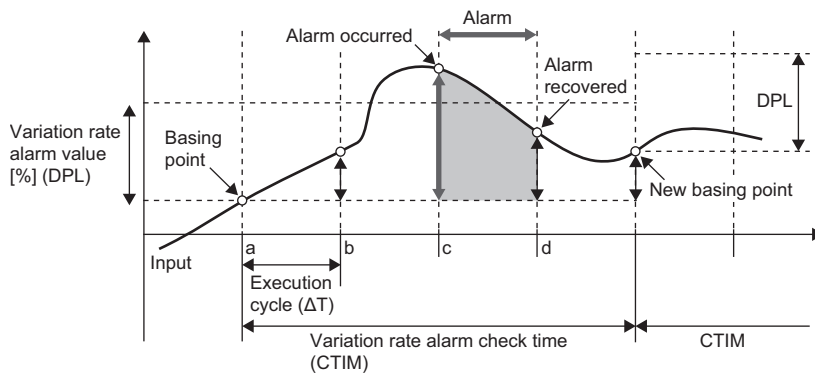
$$HH' = \frac{HH-RL}{RH-RL} \times 100(\%), PH' = \frac{PH-RL}{RH-RL} \times 100(\%)$$

$$PL' = \frac{PL-RL}{RH-RL} \times 100(\%), LL' = \frac{LL-RL}{RH-RL} \times 100(\%)$$

RH: Engineering value high limit, RL: Engineering value low limit

## Variation rate check

During the variation rate alarm check time (CTIM), this function block compares input changes with the variation rate alarm value (DPL) every execution cycle  $\Delta T$  and checks variation rate alarms.



Item		Input condition of when an alarm occurs	
		TRUE (Detected)	FALSE (Reset)
Alarm (ALM)	Positive variation rate (DPPA)	$(X_n + m) - X_n \geq DPL$	Other than the left range
	Negative variation rate (DPNA)	$(X_n + m) - X_n \leq -DPL$	Other than the left range

DPL: Variation rate alarm value (%)

m: Variation rate monitoring counter =  $CTIM/\Delta T$

$\Delta T$ : Execution cycle

CTIM: Variation rate alarm check time

$X_n$ : Reference value

## Engineering value conversion

This function block converts a process variable (%) into an engineering value.

$$PV = \frac{RH-RL}{100} \times \text{Input value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, PV: Process variable

## Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the high high limit/high limit/low limit/low low limit checks and variation rate check.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, HHA, LLA, PHA, PLA, DPPA, or DPNA of the alarm (ALM) will not be detected. <ul style="list-style-type: none"> <li>• ERR1</li> <li>• HHI</li> <li>• LLI</li> <li>• PHI</li> <li>• PLI</li> <li>• DPPI</li> <li>• DPNI</li> </ul>
Disable alarm detection by loop stop processing	Page 333 Loop stop processing

## Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) or the tag stop (TSTP) of the monitor output buffer (DOM) is TRUE, this function block performs the following processing.

- The output (PVPO) is as follows.

$$PVPO = \frac{PV-RL}{RH-RL} \times 100(\%)$$

- The control mode is automatically switched to MANUAL.
- When HHA, LLA, PHA, PLA, DPPA, or DPNA of the alarm (ALM) has occurred, the HHA, LLA, PHA, PLA, DPPA, or DPNA is reset.
- No alarm is detected in the high high limit/high limit/low limit/low low limit checks and variation rate check.

## ■ Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation			
	High high limit/high limit/low limit/low limit check	Variation rate check	Engineering value conversion	Alarm
MAN, CMV, AUT, CAS, CSV, CASDR	○	○	○	○ <sup>*1</sup>

\*1 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVPI) or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The variation rate alarm value (DPL) is less than the variation rate alarm value (DPL) in negative.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The variation rate alarm check time (CTIM) setting is less than 0.
	The value divided the variation rate alarm check time (CTIM) by the execution cycle ( $\Delta T$ ) exceeds 32767.

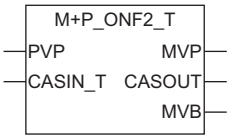


# 15.20 2 position ON/OFF Control (Enable Tracking for primary loop) (M+P\_ONF2\_T)

## M+P\_ONF2\_T

This FB performs two-position (on/off) control. The primary loop can be tracked.

### FBD/LD



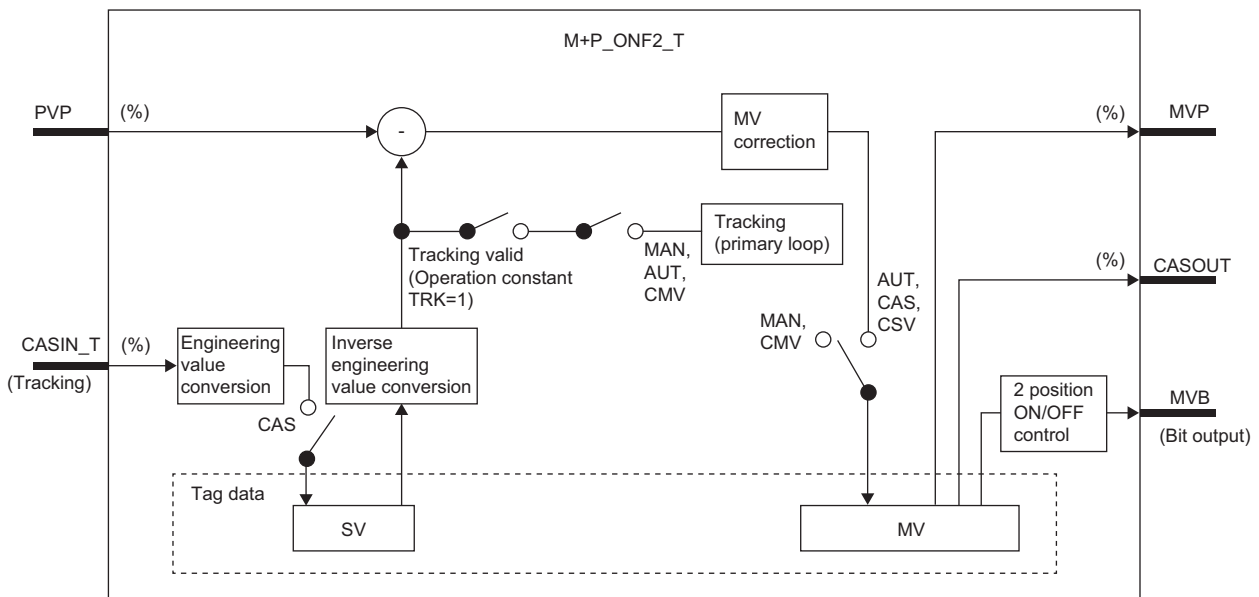
### Applicable tag type

ONF2

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN_T	Cascade SV input (unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVP	MV output (unit: %)	0 to 100[%]	Output variable	REAL
CASOUT	Cascade output (unit: %)	0 to 100[%]	Output variable	REAL
MVB	ON/OFF output (turns on at MV ≥ 50%)	TRUE: On FALSE: Off	Output variable	BOOL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
SVPTN_B0	Setting value (SV) used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SVPTN_B1	Setting value (SV) pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

## Tag data

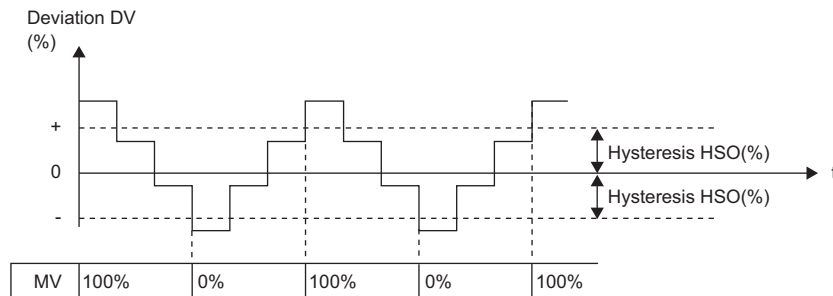
For details on tag data that is read or written with this tag access FB, refer to the following.

☞ Page 849 ONF2

## Processing details

### MV correction

This function block calculates a deviation (DV).



Condition	Deviation (DV)
Direct action (PN = 1)	DV (%) = PV (%) - SV (%)
Reverse action (PN = 0)	DV (%) = SV (%) - PV (%)

DV: Deviation (%), HSO: Hysteresis (%), MV: MV output

$$SV(\%) = \frac{SV - \text{Engineering range low limit}}{\text{Engineering range high limit} - \text{Engineering range low limit}} \times 100$$

$$PV(\%) = \frac{PV - \text{Engineering range low limit}}{\text{Engineering range high limit} - \text{Engineering range low limit}} \times 100$$

Hysteresis (%) is a percentage to (Engineering range high limit - Engineering range low limit).

### Two-position (on/off) control

This function block performs two-position (on/off) control in accordance with a manipulated value.

Condition	Two-position (on/off) output (MVB)
MV  ≥ 50 (%)	TRUE
MV  < 50 (%)	FALSE

MVB: Two-position (on/off) output, MV: MV output

Set an integral multiple of the execution cycle (ΔT) as a control cycle (CT).

### ■Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH-RL}{100} \times \text{Setting value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

### ■Inverse engineering value conversion

This function block converts the setting value (SV) of an engineering value into a setting value (SV) in percentage (%).

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

### ■Tracking processing

The following table shows whether tracking processing to the input variable CASIN\_T is performed or not.

Condition		Result
Tracking Flag (TRK)	Setting Value (SV) Used (SVPTN_B0)	
1	FALSE	The input variable CASIN_T is tracked.
	TRUE	The input variable CASIN_T is not tracked.
0	FALSE or TRUE	

### ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- Outputs (MVP) are held.
- The control mode is automatically switched to MANUAL.

### ■Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation				
	MV correction	Two-position (on/off) control	Engineering value conversion	Inverse engineering value conversion	Tracking
MAN, CMV, AUT	○	○	×	○	○*1
CAS, CSV	○	○	○	○	×

\*1 Tracking is performed when the tracking flag (TRK) is 1.

### Operation error

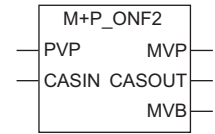
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP) or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle (ΔT) setting is less than 0.
	Hysteresis (HS0) is less than hysteresis (HS0) in negative.
	The value divided the control cycle (CT) by the execution cycle (ΔT) exceeds 32767.

# 15.21 2 position ON/OFF Control (Disable Tracking for primary loop) (M+P\_ONF2)

## M+P\_ONF2

This FB performs two-position (on/off) control.

### FBD/LD



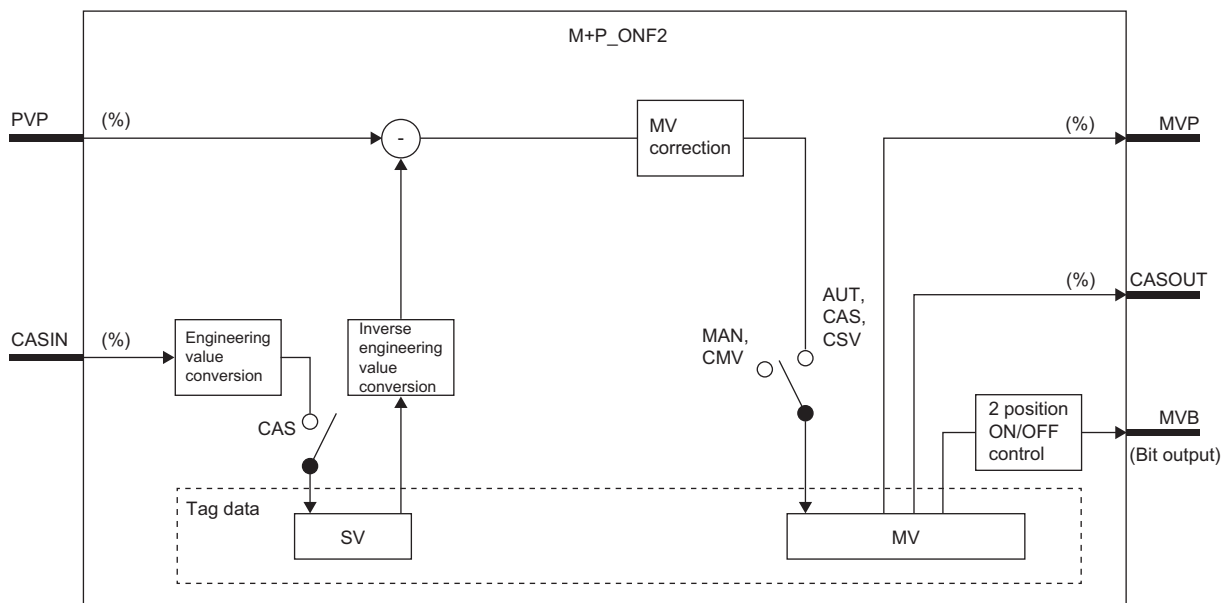
### Applicable tag type

ONF2

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



### Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN	Cascade SV input (unit: %)	0 to 100[%]	Input variable	REAL
MVP	MV output (unit: %)	0 to 100[%]	Output variable	REAL
CASOUT	Cascade output (unit: %)	0 to 100[%]	Output variable	REAL
MVB	ON/OFF output (turns on at MV ≥ 50%)	TRUE: On FALSE: Off	Output variable	REAL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
SVPTN_B0	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL

## Tag data

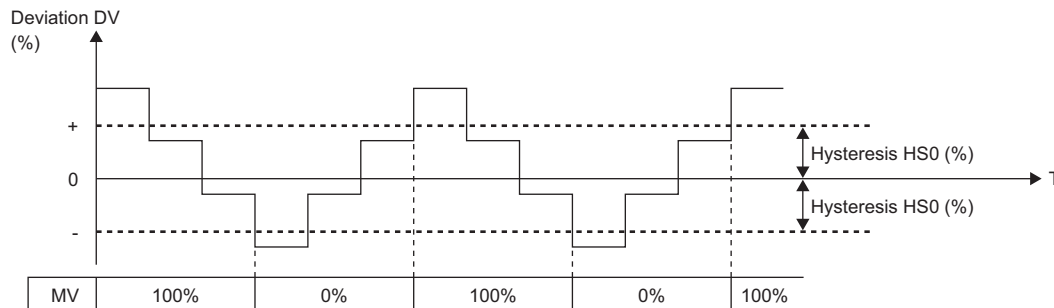
For details on tag data that is read or written with this tag access FB, refer to the following.

Page 849 ONF2

## Processing details

### MV correction

This function block calculates a deviation (DV).



Condition	Deviation (DV)
Direct action (PN = 1)	$DV (\%) = PV (\%) - SV (\%)$
Reverse action (PN = 0)	$DV (\%) = SV (\%) - PV (\%)$

DV: Deviation (%), HSO: Hysteresis (%), MV: MV output

$$SV(\%) = \frac{SV - \text{Engineering range low limit}}{\text{Engineering range high limit} - \text{Engineering range low limit}} \times 100$$

$$SV(\%) = \frac{SV - \text{Engineering range low limit}}{\text{Engineering range high limit} - \text{Engineering range low limit}} \times 100$$

Hysteresis (%) is a percentage to (Engineering range high limit - Engineering range low limit).

### Two-position (on/off) control

This function block performs two-position (on/off) control in accordance with a manipulated value.

Condition	Two-position (on/off) output (MVB)
$ MV  \geq 50 (\%)$	TRUE
$ MV  < 50 (\%)$	FALSE

MVB: Two-position (on/off) output, MV: MV output

- Set an integral multiple of the execution cycle ( $\Delta T$ ) as a control cycle (CT).

## ■Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH-RL}{100} \times \text{Setting value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Inverse engineering value conversion

This function block converts the setting value (SV) of an engineering value into a setting value (SV) in percentage (%).

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- Outputs (MVP) are held.
- The control mode is automatically switched to MANUAL.

## ■Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation			
	MV correction	Two-position (on/off) control	Engineering value conversion	Inverse engineering value conversion
MAN, CMV, AUT	○	○	×	○
CAS, CSV	○	○	○	○

## Operation error

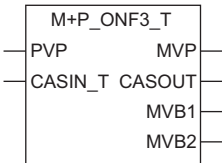
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP) or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle (ΔT) setting is less than 0.
	Hysteresis (HS0) is less than hysteresis (HS0) in negative.
	The value divided the control cycle (CT) by the execution cycle (ΔT) exceeds 32767.

# 15.22 3 position ON/OFF Control (Enable Tracking for primary loop) (M+P\_ONF3\_T)

## M+P\_ONF3\_T

This FB performs three-position (on/off) control. The primary loop can be tracked.

### FBD/LD



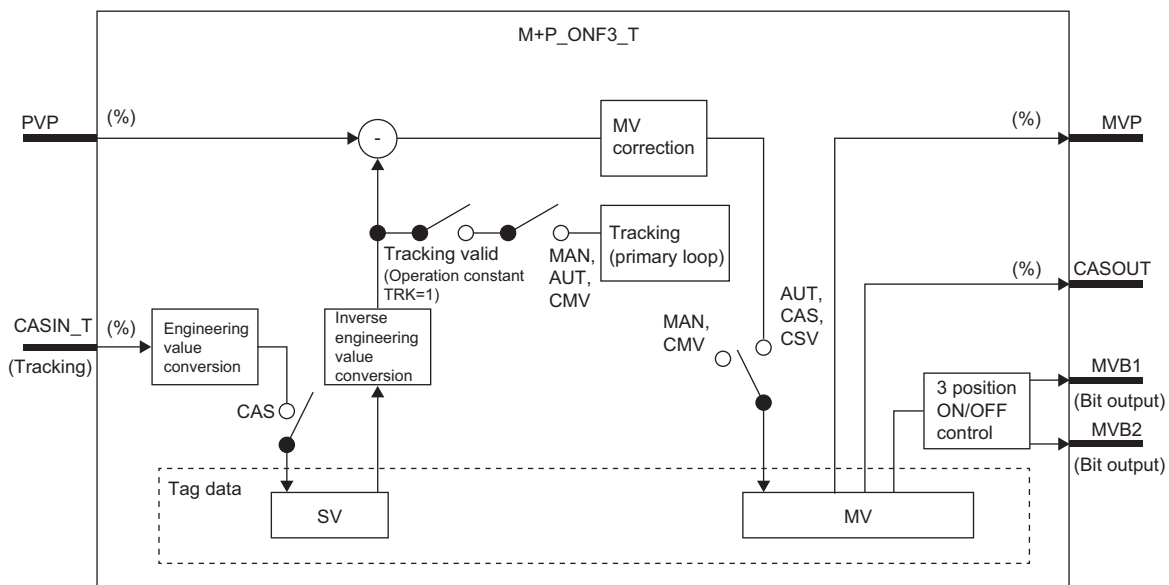
### Applicable tag type

ONF3

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



15

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN_T	Cascade SV input (unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVP	MV output (unit: %)	0 to 100[%]	Output variable	REAL
CASOUT	Cascade output (unit: %)	0 to 100[%]	Output variable	REAL
MVB1	ON/OFF output (turns on at MV ≥ 75%)	TRUE: On FALSE: Off	Output variable	BOOL
MVB1	ON/OFF output (turns on at MV < 25%)	TRUE: On FALSE: Off	Output variable	BOOL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
TRK	Tracking Flag*1	0: Not executed 1: Executed	0	User	INT
SVPTN_B0	Setting value (SV) used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SVPTN_B1	Setting value (SV) pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

## Tag data

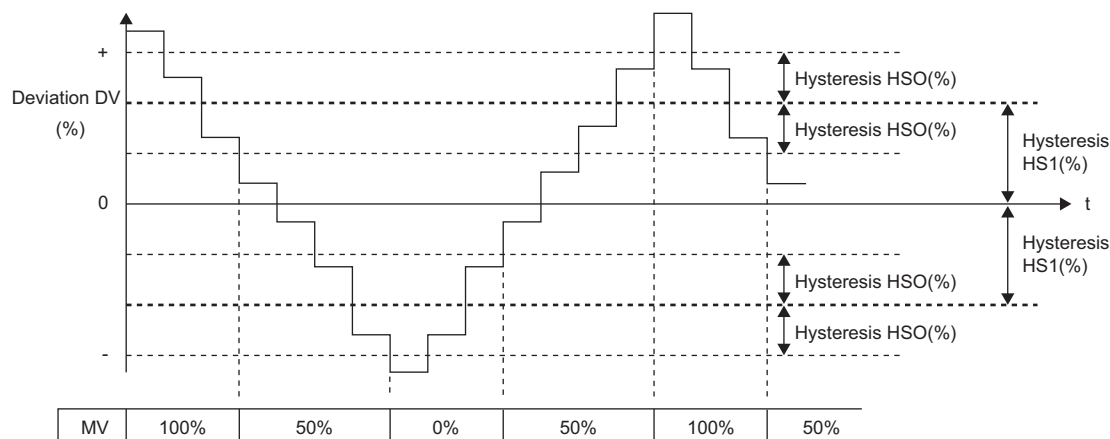
For details on tag data that is read or written with this tag access FB, refer to the following.

Page 852 ONF3

## Processing details

### MV correction

This function block calculates a deviation (DV).



Condition	Deviation (DV)
Direct action (PN = 1)	$DV (\%) = PV (\%) - SV (\%)$
Reverse action (PN = 0)	$DV (\%) = SV (\%) - PV (\%)$

DV: Deviation (%), HSO: Hysteresis (%), MV: MV output

$$SV(\%) = \frac{SV - \text{Engineering range low limit}}{\text{Engineering range high limit} - \text{Engineering range low limit}} \times 100$$

$$PV(\%) = \frac{PV - \text{Engineering range low limit}}{\text{Engineering range high limit} - \text{Engineering range low limit}} \times 100$$

Hysteresis (%) is a percentage to (Engineering range high limit - Engineering range low limit).



### ■Three-position (on/off) control

This function block performs three-position (on/off) control in accordance with a manipulated value.

Condition	Three-position (on/off) output	
	MVB1	MVB2
MV ≥ 75 (%)	TRUE	FALSE
25 (%) ≤ MV < 75 (%)	FALSE	FALSE
MV < 25 (%)	FALSE	TRUE

MVB1, MVB2: Three-position (on/off) output, MV: MV output

- Set an integral multiple of the execution cycle (ΔT) as a control cycle (CT).

### ■Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH-RL}{100} \times \text{Setting value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

### ■Inverse engineering value conversion

This function block converts the setting value (SV) of an engineering value into a setting value (SV) in percentage (%).

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

### ■Tracking processing

The following table shows whether tracking processing to the input variable CASIN\_T is performed or not.

Condition	Result	
	Tracking Flag (TRK)	Setting Value (SV) Used (SVPTN_B0)
1	FALSE	The input variable CASIN_T is tracked.
	TRUE	The input variable CASIN_T is not tracked.
0	FALSE or TRUE	

### ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- Outputs (MVP) are held.
- The control mode is automatically switched to MANUAL.

### ■Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation				
	MV correction	Three-position (on/off) control	Engineering value conversion	Inverse engineering value conversion	Tracking
MAN, CMV, AUT	○	○	×	○	○*1
CAS, CSV	○	○	○	○	×

\*1 Tracking is performed when the tracking flag (TRK) is 1.

## Operation error

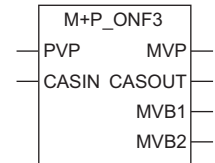
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP) or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Hysteresis 0 (HS0) is less than 0.
	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	(Hysteresis 1 (HS1) + hysteresis 0 (HS0)) is less than 0.
	Hysteresis 1 (HS1) is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.

# 15.23 3 position ON/OFF Control (Disable Tracking for primary loop) (M+P\_ONF3)

## M+P\_ONF3

This FB performs three-position (on/off) control.

### FBD/LD



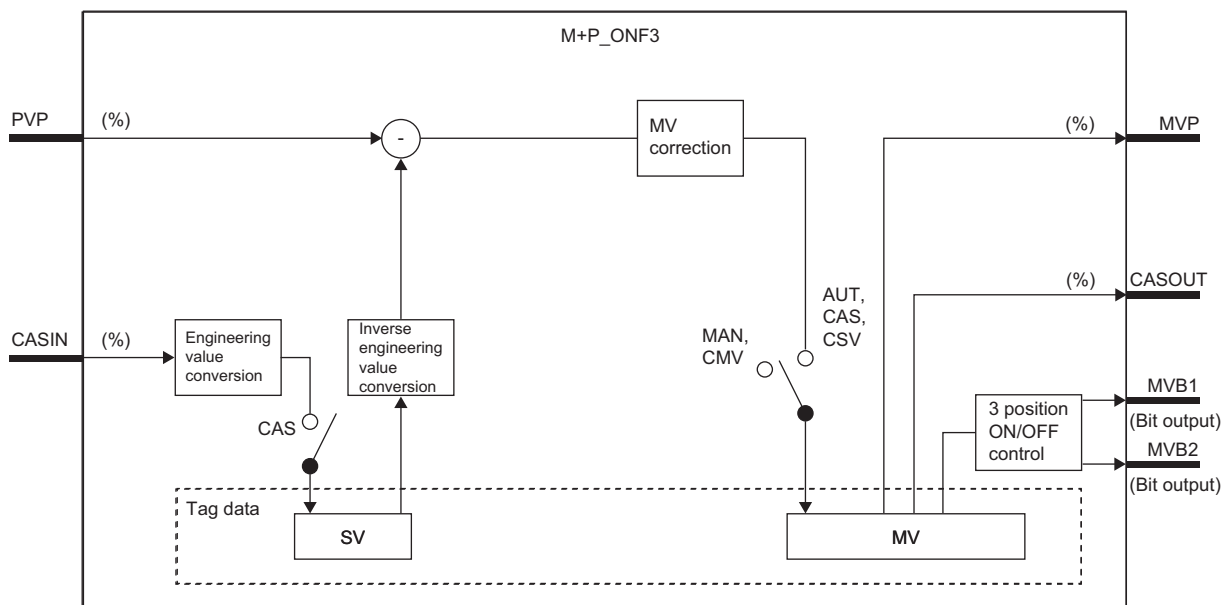
### Applicable tag type

ONF3

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



15

### Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN	Cascade SV input (unit: %)	0 to 100[%]	Input variable	REAL
MVP	MV output (unit: %)	0 to 100[%]	Output variable	REAL
CASOUT	Cascade output (unit: %)	0 to 100[%]	Output variable	REAL
MVB1	ON/OFF output (turns on at MV ≥ 75%)	TRUE: On FALSE: Off	Output variable	BOOL
MVB2	ON/OFF output (turns on at MV < 25%)	TRUE: On FALSE: Off	Output variable	BOOL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
SVPTN_B0	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL

## Tag data

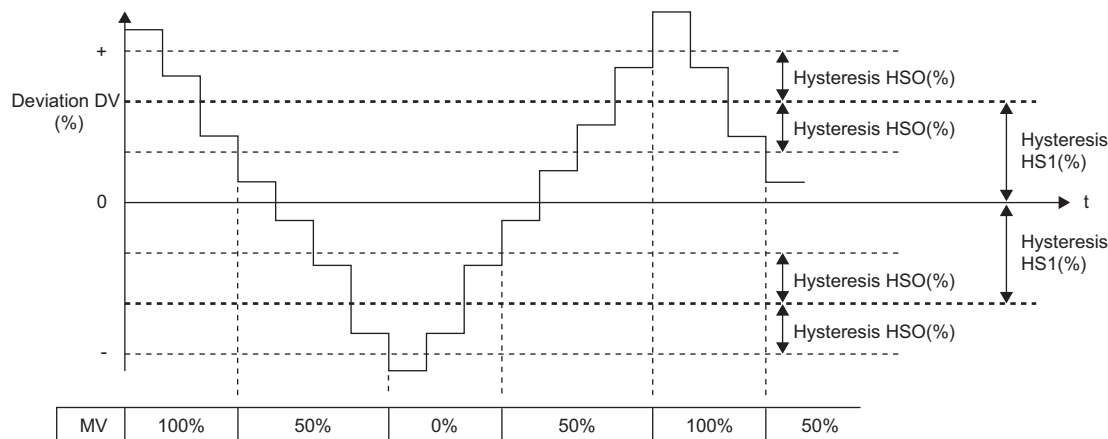
For details on tag data that is read or written with this tag access FB, refer to the following.

Page 852 ONF3

## Processing details

### MV correction

This function block calculates a deviation (DV).



Condition	Deviation (DV)
Direct action (PN = 1)	$DV (\%) = PV (\%) - SV (\%)$
Reverse action (PN = 0)	$DV (\%) = SV (\%) - PV (\%)$

DV: Deviation (%), HSO: Hysteresis (%), MV: MV output

$$SV(\%) = \frac{SV - \text{Engineering range low limit}}{\text{Engineering range high limit} - \text{Engineering range low limit}} \times 100$$

$$PV(\%) = \frac{PV - \text{Engineering range low limit}}{\text{Engineering range high limit} - \text{Engineering range low limit}} \times 100$$

Hysteresis (%) is a percentage to (Engineering range high limit - Engineering range low limit).

### Three-position (on/off) control

This function block performs three-position (on/off) control in accordance with a manipulated value.

Condition	Three-position (on/off) output	
	MVB1	MVB2
$MV \geq 75 (\%)$	TRUE	FALSE
$25 (\%) \leq MV < 75 (\%)$	FALSE	FALSE
$MV < 25 (\%)$	FALSE	TRUE

MVB1, MVB2: Three-position (on/off) output, MV: MV output

- Set an integral multiple of the execution cycle ( $\Delta T$ ) as a control cycle (CT).

## ■Engineering value conversion

This function block converts the setting value (%) from the primary loop in the CAS or CSV mode into an engineering value.

$$SV = \frac{RH-RL}{100} \times \text{Setting value (\%)} + RL$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Inverse engineering value conversion

This function block converts the setting value (SV) of an engineering value into a setting value (SV) in percentage (%).

$$SV(\%) = \frac{100}{RH-RL} \times (SV-RL)$$

RH: Engineering value high limit, RL: Engineering value low limit, SV: Setting value

## ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- Outputs (MVP) are held.
- The control mode is automatically switched to MANUAL.

## ■Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation			
	MV correction	Three-position (on/off) control	Engineering value conversion	Inverse engineering value conversion
MAN, CMV, AUT	○	○	×	○
CAS, CSV	○	○	○	○

15

## Operation error

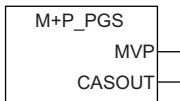
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (PVP) or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Hysteresis 0 (HS0) is less than 0.
	The control cycle (CT) setting is less than 0.
	The execution cycle (ΔT) setting is less than 0.
	(Hysteresis 1 (HS1) + hysteresis 0 (HS0)) is less than 0.
	Hysteresis 1 (HS1) is less than 0.
	The value divided the control cycle (CT) by the execution cycle (ΔT) exceeds 32767.

# 15.24 Program Setter (M+P\_PGS)

## M+P\_PGS

This FB registers time width sets and setting value programs of up to 16 steps, and outputs a setting value corresponding to the progress time of each step with linear interpolation. ( Page 937 Program setter (PGS) )

### FBD/LD



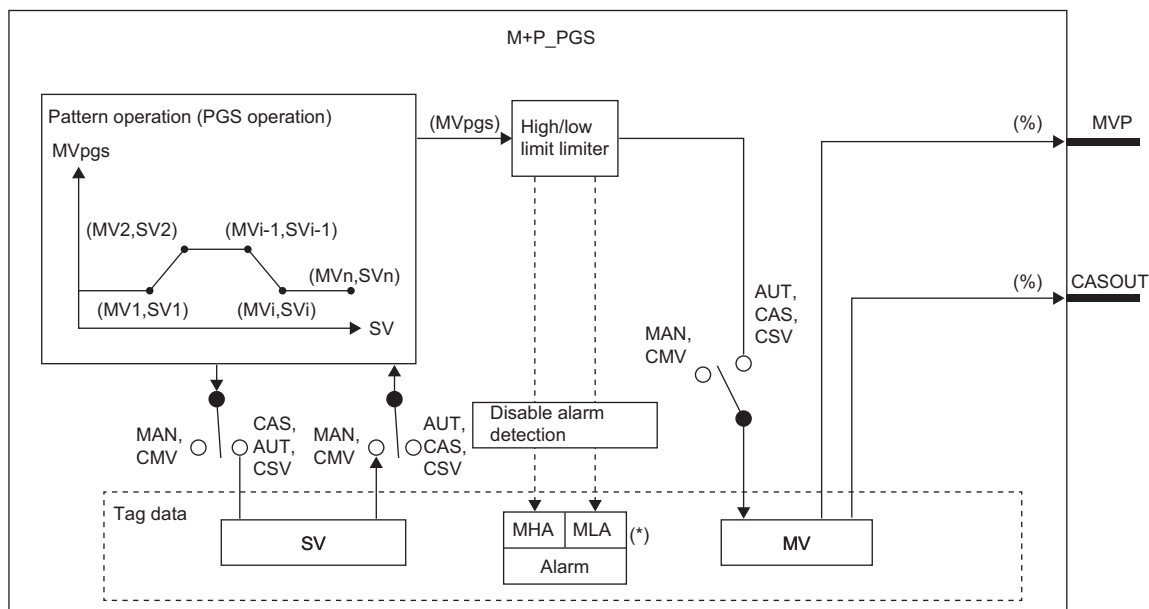
### Applicable tag type

PGS

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bit item.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
MVP	MV output (unit: %)	0 to 100[%]	Output variable	REAL
CASOUT	Cascade output (unit: %)	0 to 100[%]	Output variable	REAL

#### Tag data

For details on tag data that is read or written with this tag access FB, refer to the following.

Page 869 PGS

## Processing details

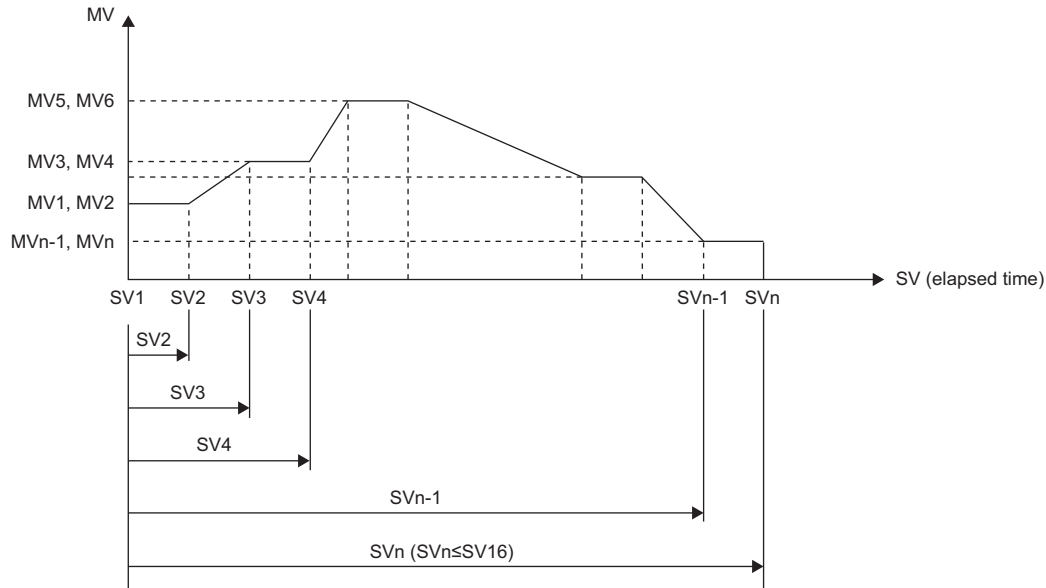
### ■PGS operation

This function block outputs predetermined values as time proceeds.

There are the following three operation types: hold type, return type, and cyclic type.

Operation type	Description
Hold	Outputs data while holding the value of setting time (SVn).
Return	Sets the set value (SV) to 0 and outputs the last value of the manipulated value (MV).
Cyclic	Processes based on the setting time 1 (SV1) to setting time n (SVn) and then restarts processing from setting time 1 (SV1).

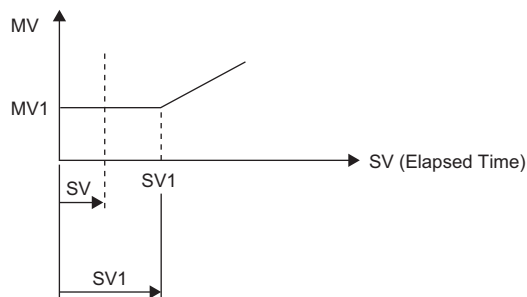
When the stop alarm (SPA) of the alarm (ALM) is FALSE, this function block performs the following processing. (loop run processing).



SV (elapsed time) is in units of seconds.

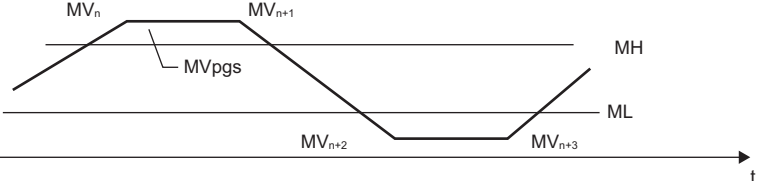
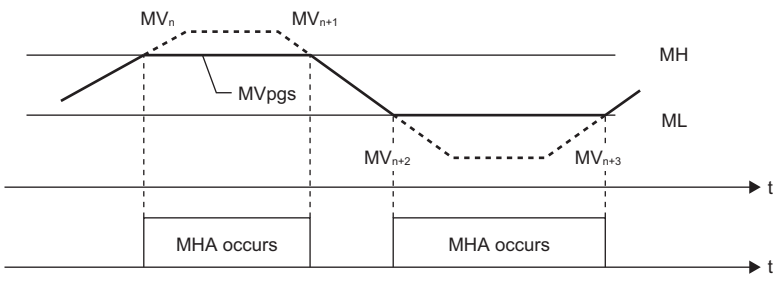
Type		Hold	Return	Cyclic
<b>Control mode</b>		<b>AUT</b>	<b>AUT</b>	<b>CAS, CSV</b>
MVpgs calculation	Elapsed time < SV1	MV1		
	$SV_i - 1 \leq \text{Elapsed time} < SV_i$	$\frac{MV_i - MV_{i-1}}{SV_i - SV_{i-1}} \times (\text{Elapsed time} - SV_{i-1}) + MV_{i-1}$		
Elapsed time > SVn <sup>*1</sup>	Control mode transition	MANUAL	MANUAL	No transition
	SV (elapsed time)	Last value (SVn)	0	0
	MV	Last value (MVn)	Last value (MVn)	MV1
	Restart	MAN → AUT operation after setting SV (elapsed time)	MAN → AUT operation	Automatic restart

\*1 The following figure shows the result of when SV1 is longer than 0 second.



## High/low limit check


This function block performs the high/low limit checks.

Control mode	Description
MAN, CMV	 <ul style="list-style-type: none"> <li>• MVpgs is not fixed with MH or ML even though MVn is equal to or greater than MH or MVn is equal to or smaller than ML.</li> <li>• No MHA or MLA of the alarm (ALM) is detected even though MVpgs is equal to or greater than MH or MVpgs is equal to or smaller than ML.</li> </ul>
AUT, CAS, CSV	 <ul style="list-style-type: none"> <li>• MVpgs is fixed with MH or ML when MVn is equal to or greater than MH or MVn is equal to or smaller than ML. (When MVn has been programmed like the broken lines in the above diagram, MVpgs will be fixed as the solid lines in the above diagram and output.)</li> <li>• MHA or MLA of the alarm (ALM) will occur when MVn is equal to or greater than MH or MVn is equal to or smaller than ML.</li> </ul>

MVn: MV output  
 MH: Output high limit value  
 ML: Output low limit value  
 MHA: Output high limit alarm  
 MLA: Output low limit alarm

## Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the high/low limiter processing.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, MHA or MLA of the alarm (ALM) will not be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• MHI</li> <li>• MLI</li> </ul>
Disable alarm detection by control mode selection	In the MAN or CMV mode, MHA and MLA of the alarm (ALM) are reset and MHA or MLA will not be detected.
Disable alarm detection by loop stop processing	 Page 350 Loop stop processing

## Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- Outputs (MVP) are held.
- The control mode is automatically switched to MANUAL.
- When MHA or MLA of the alarm (ALM) has occurred, the MHA or MLA is reset.
- No alarm is detected in the variation rate & high/low limiter processing.

### Point

Even when an operation constant "Number of points" (PTNO) of tag data is 0, the same processing as loop stop processing is performed.



## ■ Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation		
	PGS pattern operation	High/low limit check	Alarm
MAN*1, CMV	×	×	×*2
AUT, CAS, CSV	○	○	○*3

\*1 When an operation constant "Number of points" (PTNO) is 0, the same processing as loop stop processing is performed and the control mode will be switched to MANUAL.

\*2 An alarm (ALM) whose corresponding bit is TRUE (detected) is reset, and the alarm will not be detected.


\*3 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

## Operation error

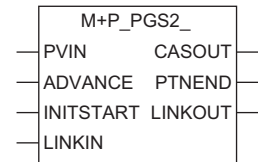
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The operation constant "Number of points" (PTNO) is less than 0 or greater than 16.

# 15.25 Multi-Point Program Setter (M+P\_PGS2\_)

## M+P\_PGS2\_

This FB registers time width sets and setting value programs of up to 32 steps, and outputs a setting value corresponding to the progress time of each step with linear interpolation. (  Page 938 Multi-point program setter (PGS2))

### FBD/LD



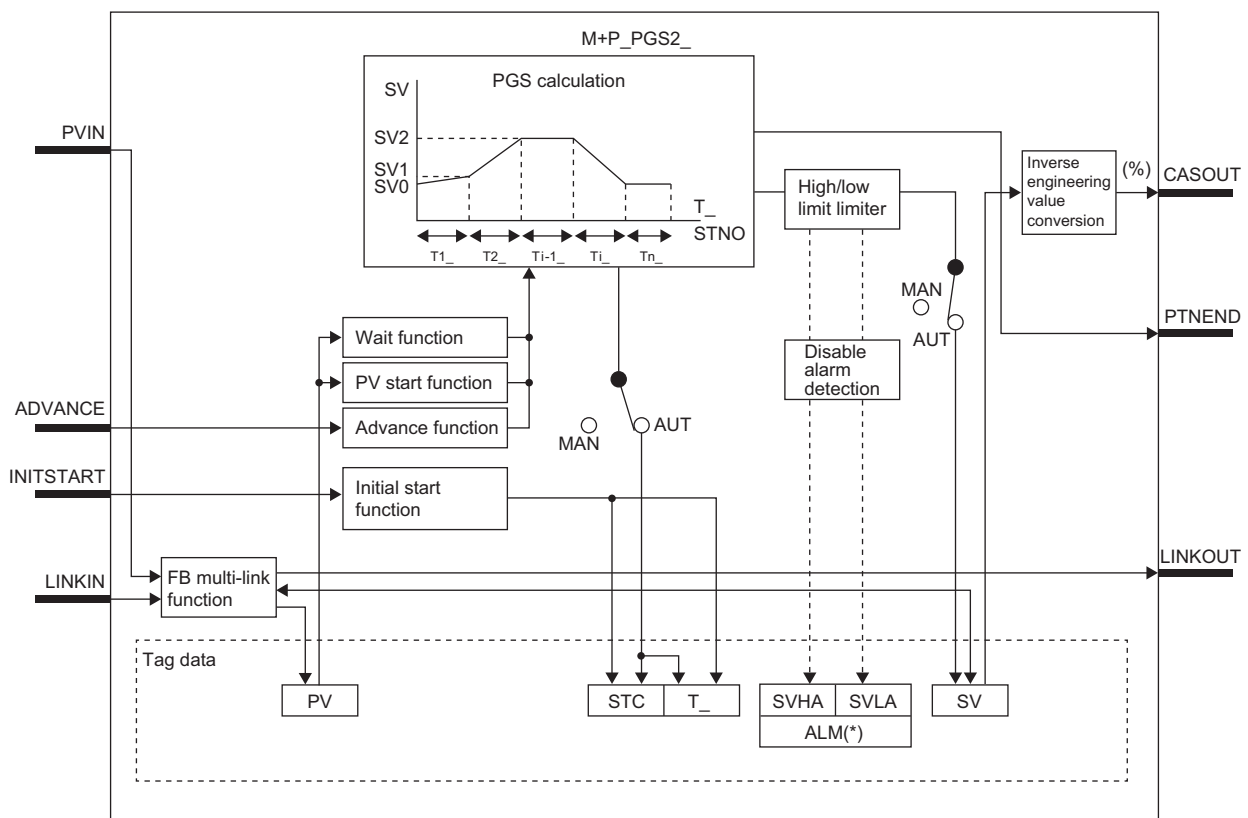
### Applicable tag type

PGS2

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	—	—	—

### Block diagram



## Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVIN	Process input (Engineering value)	-32768 to 32767	Input variable	REAL
ADVANCE	Advance command	TRUE, FALSE	Input variable	BOOL
INITSTART	Initial start command	TRUE, FALSE	Input variable	BOOL
LINKIN	Link input (Indirect address)	—	Input variable	DWORD
CASOUT	Cascade output (unit: %)	0 to 100[%]	Output variable	REAL
PTNEND	Pattern end output	TRUE, FALSE	Output variable	BOOL
LINKOUT	Link output (Indirect address)	—	Output variable	DWORD

### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
PVSTARTNO	PV Start Search Start Step	1 to 32	1	User	INT
PVENDNO	PV Start Search End Step	1 to 32	32	User	INT
PRIMARY	Lead FB specified	TRUE: Lead FALSE: Following	TRUE	User	BOOL


### ■Public variable (others) \*1

Variable name	Description	Recommended range	Initial value	Set by	Data type
TCNT	Second counter for minute mode.	0 to 59	0	System	INT
TMCNT	Millisecond counter for second mode.	0 to 999	0	System	INT

\*1 Read or write the variables using a program. They are not displayed in "FB Property" of the engineering tool.

### ■Tag data

For details on tag data that is read or written with this tag access FB, refer to the following.

 Page 872 PGS2

## Processing details

### ■PV start

At the start of a control (when the AUT mode is switched), this function block reduces a gap with process variable by referring to the process variable and adjusting the control start point. The adjusted control start point is set as the start point (current) (SV0C).

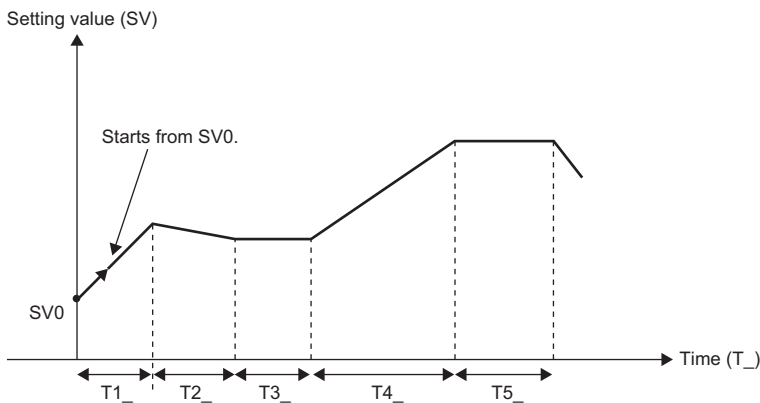
PV start type	Control start condition				
	STNO = 0	STNO ≠ 0			After the second cycle at cyclic
		STC = 0 and T_ = 0	STC = 0 and T_ ≠ 0	STC ≠ 0	
PVSTART = 0	(1)	SV0 start (2)	(5)	(6)	(7)
PVSTART = 1		PV start 1 (3)			
PVSTART = 2		PV start 2 (4)			

- (1) When the number of steps (STNO) is 0

Because the step setting is invalid, the control mode will be immediately switched to the MAN mode, if the mode has been switched to the AUT mode, and the pattern completion output (PTNEND) will turn on by only one cycle. The system operates in the MAN mode, and the FB multi-link function and inverse engineering value conversion are activated.

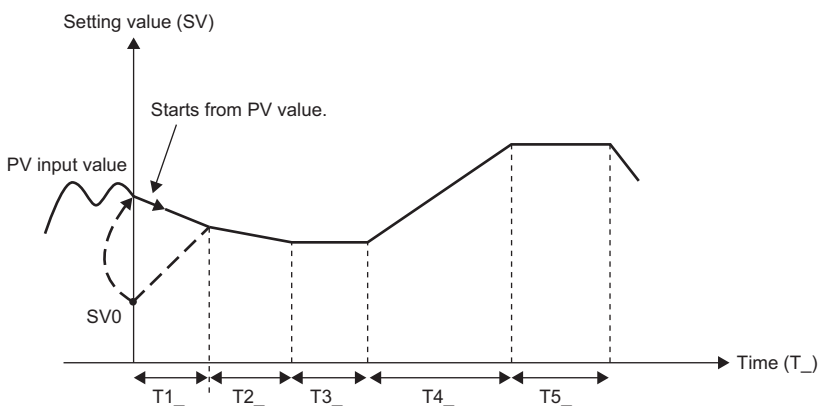
- (2) SV0 start (Fixed start point: PVSTART = 0)

The control will start using the start point (SV0) as the control start point without referring to the process variable.



- (3) PV start 1 (Start point correction: PVSTART = 1)

The control will start using the process variable as the control start point.



- (4) PV start 2 (Start point searching: PVSTART = 2)

The process variable and the PV start search start step (PVSTARTNO) are used to search a match point within the range specified by the PV start search end step (PVENDNO), and the control will start with the step number or time.

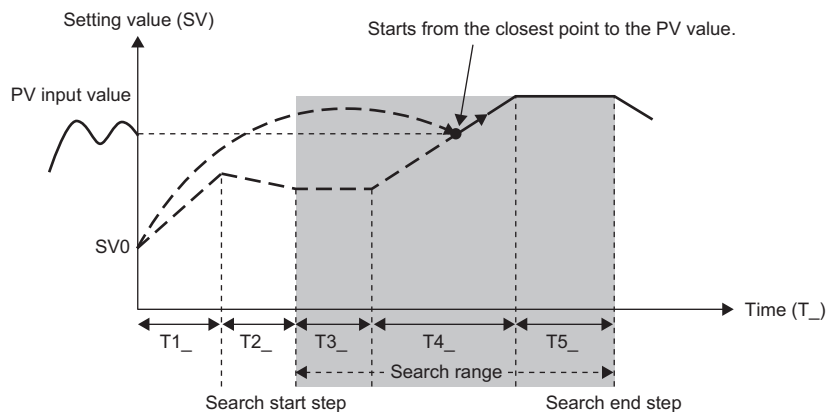
When no match point is found, the closest point (maximum point or minimum point) is used.

If multiple maximum points or minimum points are found, the point that comes last is used.

When a match point is the end point of the last step, the control will be immediately completed and the setting value before the mode switching to the AUT mode will be held.

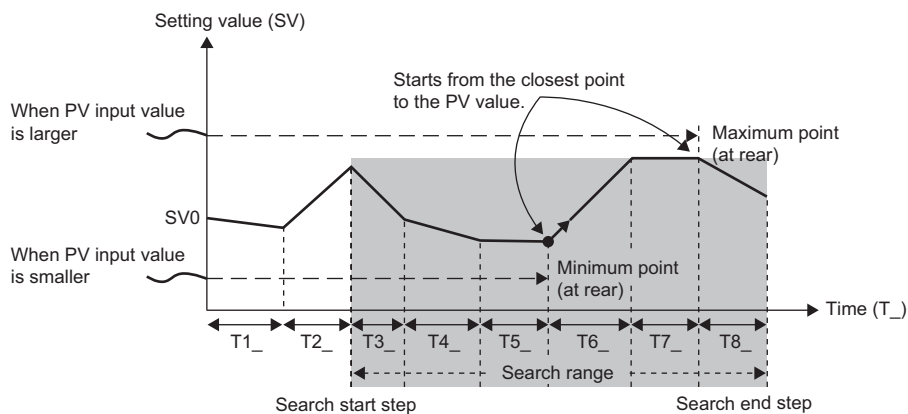
**Ex.**

When a match point was found within the search range



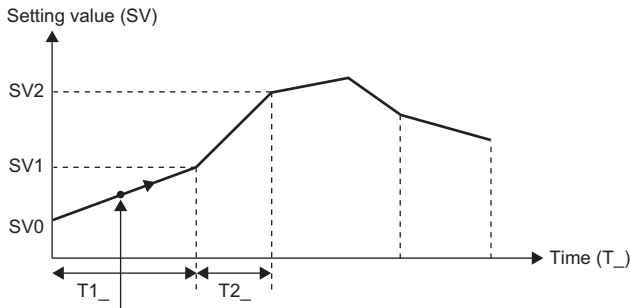
**Ex.**

When no match point was found within the search range



- (5) Starting a control when  $STC = 0$  and  $T_ \neq 0$

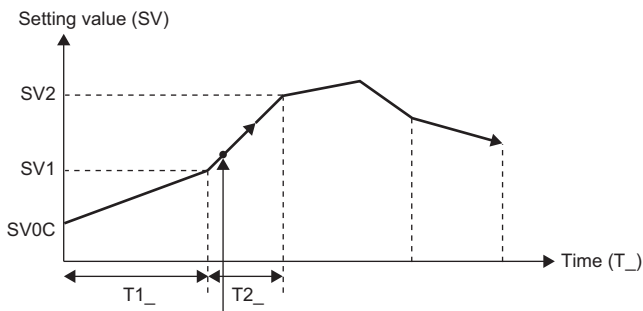
The control will start from the current time ( $T_$ ) at  $STC = 1$  using the start point ( $SV_0$ ) as the control start point.



Starts from the current time ( $T_$ ) at  $STC=1$ .

- (6) Starting a control when  $STC \neq 0$

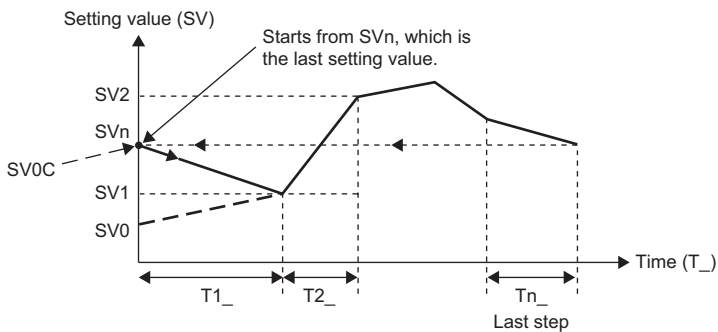
The control will start from the current  $STC$  and  $T_$  using the control start point (start point (current) ( $SV_{0C}$ )) of the previous control (AUT mode).



Starts from the current  $STC$  and  $T_$ .

- (7) After the second cycle at cyclic

The control will start using the last setting value ( $SV_n$ ) as the next control start point (start point (current) ( $SV_{0C}$ )).



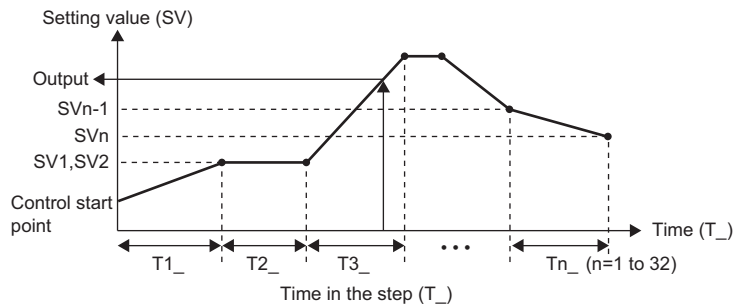
## PGS calculation

In the AUT mode, this function block outputs setting values to the time predetermined by each step as time proceeds.

There are the following three operation types: hold type, return type, and cyclic type.

Time width sets and setting values of up to 32 steps can be registered in the format of (Tn\_, SVn).

The control start point depends on the PV start type (PVSTART) setting.



- Step specification execution

In the AUT mode, changing the executing step number (STC) jumps to the start of a specified step (T\_ = 0).

Changing the time in the step (T\_) jumps to the time of the step.

To jump to the end in the step (T = Tn) by changing the time in the step (T\_) when the wait function is valid, refer to the wait function.

- Step management

Progress processing of the time in the step (T\_) and the executing step number (STC) is performed in the AUT mode.

Condition	Processing result	
	Time in the step (T_)	Executing Step No. (STC)
STC ≤ 0	0	1
STC > 0	T_ < 0	Last value
	MV < 0 ≤ T_ < Ti_	T_ + ΔT <sup>*1</sup>
	Ti_ ≤ T_ <sup>*2</sup>	Move to the next step (STC + 1)

ΔT: Execution cycle, i: Executing step number (STC)

\*1 The execution cycle addition of the time in the step (T\_) is calculated in real numbers with a resolution in units of milliseconds when the second has been specified for the unit of time (TUNIT), or with a resolution in units of seconds when the minute has been specified for the unit of time (TUNIT).

\*2 To enable and use the wait function, refer to the wait function.

- SV<sub>PGS</sub> calculation

In the AUT mode, a setting value is calculated according to the executing step number (STC) and the time in the step (T\_).

Condition	Processing result
Ti_ ≤ 0	SV <sub>i</sub>
0 < T_ ≤ Ti_	$\frac{SV_i - SV_{i-1}}{Ti\_} \times T\_ + SV_{i-1}$ (SV <sub>PGS</sub> is calculated in real numbers.)

i: Executing step number (STC)

- Processing of when the last step processing is completed

In the AUT mode, the pattern end output (PTNEND) is turned on by one cycle and the following processing is performed after patterns are performed and the last step is ended.

Item	Operation Type		
	Hold	Return	Cyclic
Control mode transition	Transition to MAN	Transition to MAN	No transition
Time in the step (T <sub>n</sub> )	Last value	0	0
Executing Step No. (STC)	Last value	0	1
SV <sub>PGS</sub> calculation	SV output value of the last step (SV <sub>n</sub> )	SV output value of the last step (SV <sub>n</sub> )	Processing is restarted from step 1 with the SV output value of the last step (SV <sub>n</sub> ) as the control start point.

### Point

- Setting the time width T1<sub>n</sub> to Tn<sub>n</sub> in PGS calculation

Time width is set in integer number. The number of seconds or minutes is set in a loop tag.

The unit of time is common to all steps and is specified by the unit of time (TUNIT) of tag data.

The maximum value of the time width setting is 32767 seconds (approximately 9 hours) or 32767 minutes (approximately 22 days) for each step.

- Setting the setting value SV1 to SV<sub>n</sub> in PGS calculation

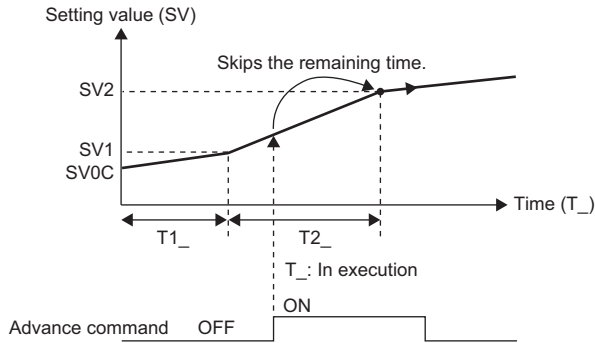
Set setting values in engineering values. The setting range is -32768 to 32767 and set the values as engineering values. A value with a decimal point cannot be specified.

When a value with a decimal point needs to be set depending on the engineering value range, when 1.5MPa is set, for example, convert the value into 1500kPa to fit the value within the range of -32768 to 32767.



### ■Advance function

In the AUT mode, this function block proceeds to the next step by forcibly ending the step in execution. This function is executed on the rising edge of the input variable ADVANCE command and proceeds the processing by one step. To proceed the processing by one step next time, turn off the command and then turn it on again.



When the advance command is executed in the waiting status, the status is cleared and the processing shifts to the one of the next step.

When the advance command is executed at the last step, the control will be immediately completed and the setting value before the execution of the advance command will be held.

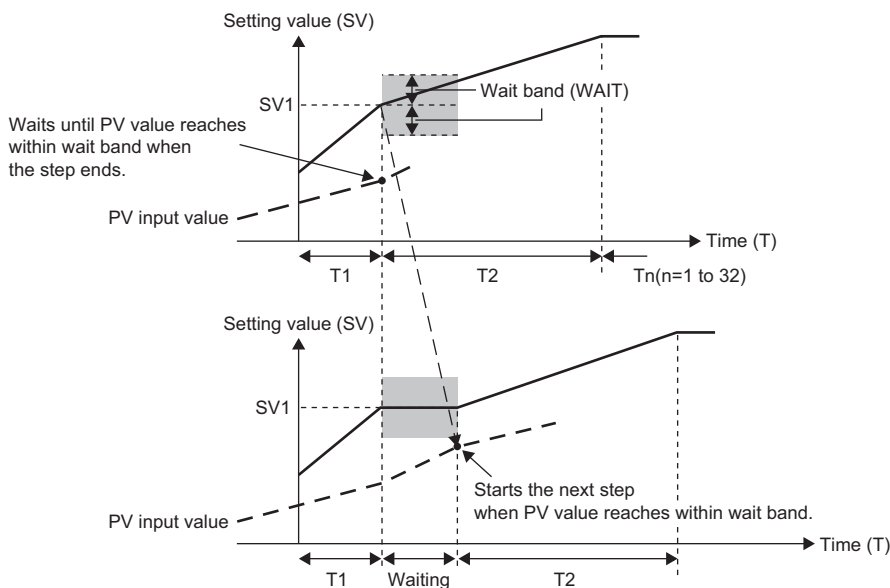
### ■Wait function

In the AUT mode, this function block checks if the process variable (PV) has followed the setting value (SV), and controls the progress of steps every time each step is completed.

The wait width setting is common to all steps.

Condition			Processing result
Control mode	Wait Width	Process Variable	
MAN	—	—	The wait function is not executed.
AUT	WAIT ≤ 0	—	
	WAIT > 0	PV - SV  > WAIT	Transition to the next step stops. The last value of the step is held as the setting value (SV). <sup>*1</sup>
		PV - SV  ≤ WAIT	Transition to the next step is performed.

\*1 When the time in the step was changed and the step was completed (T = T<sub>n</sub>), the setting value (SVPGS) immediately before the change of the time in the step is held instead of the last value of the step. To hold the last value of the step (SV<sub>n</sub>), change the time in the step to the time immediately before the end of the step (T = T<sub>n</sub> - 1).



## ■Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the setting value high/low limiter processing.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, SVHA or SVLA of the alarm (ALM) will not be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• SVHI</li> <li>• SVLI</li> </ul>
Disable alarm detection by loop stop processing	Page 362 Loop stop processing

## ■Setting value high/low limiter

In the AUT mode, this function block checks the high/low limits.

Condition	Setting Value (SV)	Alarm (ALM)	
		SV low limit (SVLA)	SV high limit (SVHA)
MAN	SV <sub>PGS</sub>	FALSE (Reset)	FALSE (Reset)
AUT	SV <sub>PGS</sub> > SH	FALSE (Reset)	TRUE (Detected)
	SV <sub>PGS</sub> < SL	TRUE (Detected)	FALSE (Reset)
	SL ≤ SV <sub>PGS</sub> ≤ SH	FALSE (Reset)	FALSE (Reset)

Control mode	Description
MAN	
AUT	

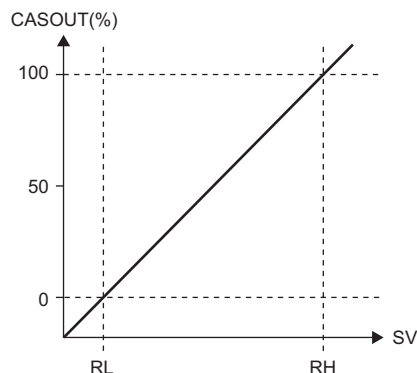
SH: SV high limit value  
SL: SV low limit value  
SVHA: SV high limit alarm  
SVLA: SV low limit alarm

### ■Inverse engineering value conversion

This function block converts the setting value (SV) into a value in percentage (%) within the engineering value range of RL to RH.

$$CASOUT(\%) = \frac{SV-RL}{RH-RL} \times 100$$

CASOUT: Cascade output  
 RH: Engineering value high limit  
 RL: Engineering value low limit  
 SV: Setting value



### ■Initial start function

This function block initializes the executing step number (STC) and the time in the step (T\_) on the rising edge of an input variable INITSTART command, and switches the control mode to the AUT mode when the module is in MAN mode.

Type	Variable name	Data type	Description	Value to be initialized
Loop tag	STC	INT	Executing Step No.	0
	T_	INT	Time in the step	0
Public variable	TCNT	INT	Second counter for minute mode.	0
	TMCNT	INT	Millisecond counter for second mode.	0

### ■FB multi-link function

To create a program having over 32 steps, link multiple tag FBs of the M+M\_PGS2\_ type or user-defined tag FBs of the PGS2 type so that they function as a program setter, and perform the following operations.

- Use a common setting value for the multiple FBs so that the same setting value will be the last output value even after the setting value of an FB has changed.
- The control mode of the following FB is preferentially managed, so that multiple FBs are not switched to the AUT mode when the AUT mode is switched.
- The process variable input to the start FB is copied to the process variable of the following FB.

For programs having multi-linked FBs, refer to the following.

☞ Page 494 M+M\_PGS2\_

## ■ Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) or the tag stop (TSTP) of the monitor output buffer (DOM) is TRUE, this function block performs loop stop processing.

Loop stop processing result					
Input (PV)	Executing Step No. (STC)/Time in the step (T_)	Output (SV)	Mode	Alarm reset <sup>*1</sup>	Alarm detected <sup>*2</sup>
Hold	Hold	Hold	MAN	Reset	Not detected

\*1 When SVLA or SVHA of the alarm (ALM) has occurred, the SVLA or SVHA is reset.

\*2 No alarm is detected in the setting value high/low limiter processing.

## ■ Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation				
	PV start	PGS calculation	Advance function	Wait function	Alarm
MAN	×	×	×	×	×
AUT	○	○	○	○	○ <sup>*1</sup>

Control mode	Processing operation			
	Setting value high/low limiter	Inverse engineering value conversion	Initial start function	FB multi-link function
MAN	×	○	○	○
AUT	○	○	○	○

\*1 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

## Operation error

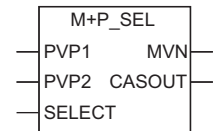
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data or operation result within a function block is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# 15.26 Loop Selector (Disable Tracking for primary loop) (M+P\_SEL)

## M+P\_SEL

This FB selects an input value depending on the setting of the select signal and outputs the result. The input 1 and 2 are not tracked.

### FBD/LD



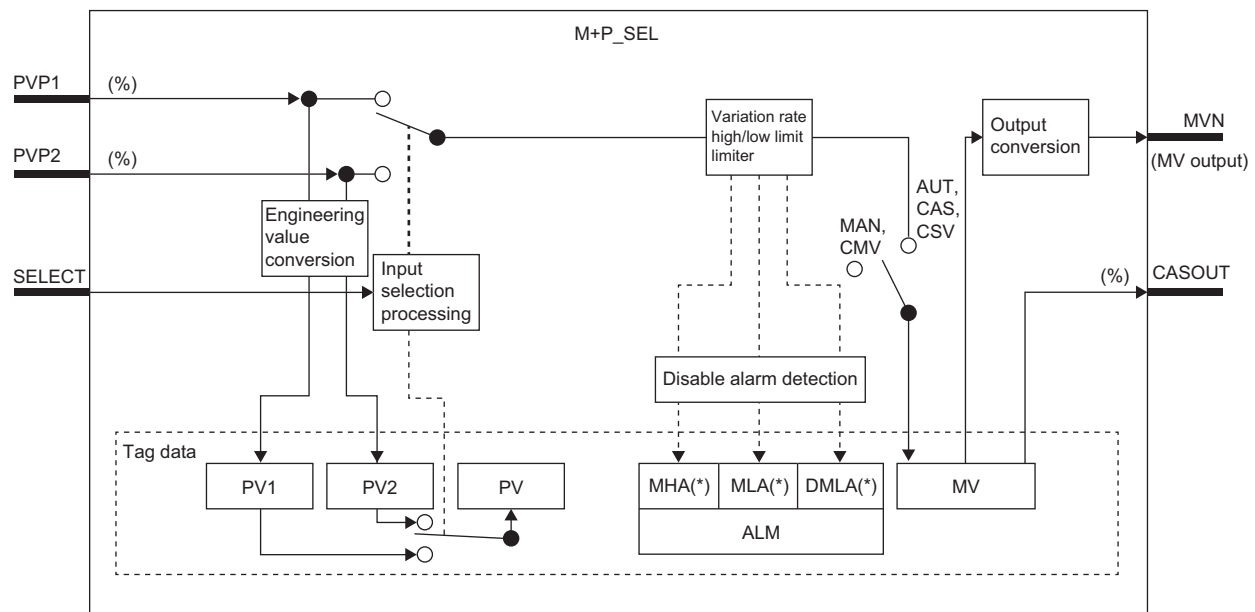
### Applicable tag type

SEL

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bit item.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP1	PV input (unit: %)	0 to 100[%]	Input variable	REAL
PVP2	PV input (unit: %)	0 to 100[%]	Input variable	REAL
SELECT	Selection signal	TRUE: PVP2 FALSE: PVP1	Input variable	BOOL
MVN	Output to a module	NMIN to NMAX	Output variable	REAL
CASOUT	Cascade output (unit: %)	0 to 100[%]	Output variable	REAL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL

## Tag data

For details on tag data that is read or written with this tag access FB, refer to the following.

☞ Page 866 SEL

## Processing details

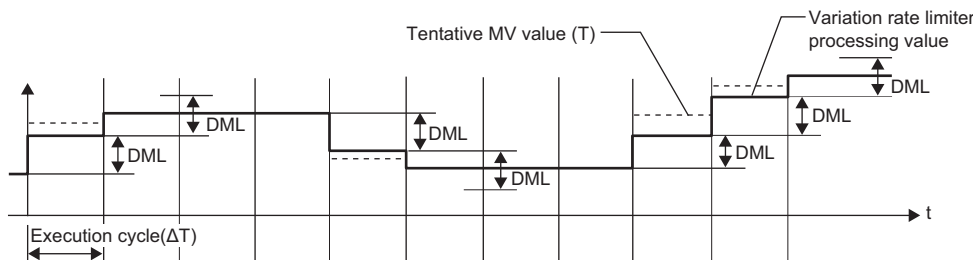
### Input selection processing

Selection signal (SELECT)	Result
FALSE	The input PVP1 (%) is selected.
TRUE	The input PVP2 (%) is selected.

### Variation rate & high/low limiter

This function block checks the variation rate and high/low limits of the input value.

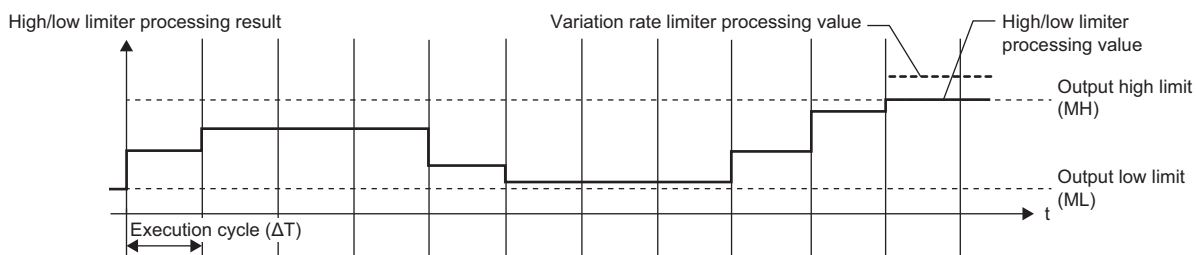
- Variation rate limiter



Condition	Variation rate limiter processing result	Alarm (ALM)
		Output variation rate limit (DMLA)
$ T - MV  \leq DML$	T	FALSE (Reset)
$T - MV > DML$	$MV + DML$	TRUE (Detected)
$T - MV < -DML$	$MV - DML$	TRUE (Detected)

T: Tentative manipulated value, MV: Manipulated value, DML: Output variation rate high limit value

- High/low limiter



Condition	High/low limiter processing result	Alarm (ALM)	
		Output Low Limit Alarm (MLA)	Output High Limit Alarm (MHA)
Variation rate limiter processing result $> MH$	MH	FALSE (Reset)	TRUE (Detected)
Variation rate limiter processing result $< ML$	ML	TRUE (Detected)	FALSE (Reset)
$ML \leq$ Variation rate limiter processing result $\leq MH$	Variation rate limiter processing value	TRUE (Detected)	FALSE (Reset)

MH: Output high limit value, ML: Output low limit value

### Engineering value conversion

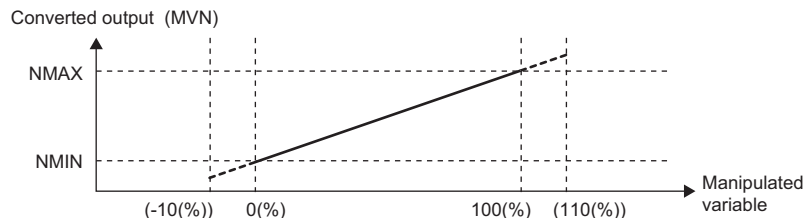
This function block converts an input value (%) to an engineering value.

$$PVn = \frac{RH-RL}{100} \times PVPn(\%) + RL$$

- RH: Engineering value high limit
- RL: Engineering value low limit
- PVPn: PV input value (%)
- PVn: PV1, PV2

### Output conversion

This function block performs output conversion processing.



$$\text{Converted output (MVN)} = \left\{ (NMAX - NMIN) \times \frac{MV}{100} \right\} + NMIN$$

- NMAX: Output conversion high limit value
- NMIN: Output conversion low limit
- MV: Manipulated value (%)
- MVN: Output conversion output value

### Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the variation rate & high/low limiter processing.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no DMLA, MHA, or MLA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DMLI</li> <li>• MHI</li> <li>• MLI</li> </ul>
Disable alarm detection by loop stop processing	Page 365 Loop stop processing

### Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- Outputs (MVN) are held.
- The control mode is automatically switched to MANUAL.
- When DMLA, MHA, or MLA of the alarm (ALM) has occurred, the DMLA, MHA, or MLA is reset.
- No alarm is detected in the variation rate & high/low limiter processing.

### Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation			
	Engineering value conversion	Variation rate & high/low limiter	Output conversion	Alarm
MAN, CMV	○	×	○	○*1
AUT, CAS, CSV	○	○	○	○*1

\*1 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

### Operation error

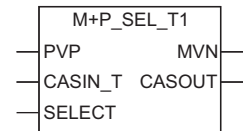
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data 1 (PVP1), operation constant (NMAX, NMIN), tag data, or input data 2 (PVP2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# 15.27 Loop Selector (Enable Tracking for primary loop only by CASIN\_T) (M+P\_SEL\_T1)

## M+P\_SEL\_T1

This FB selects an input value depending on the setting of the select signal and outputs the result. The input 1 is not tracked. The primary loop of the input 2 is tracked, but tracking from the secondary loop is not performed.

### FBD/LD



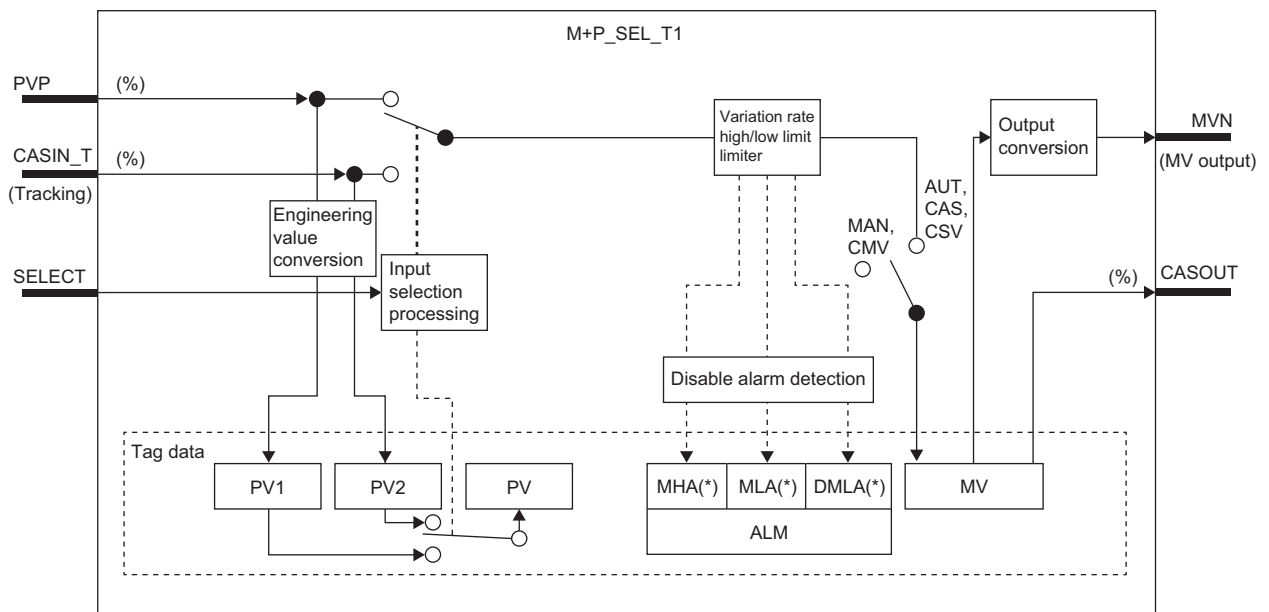
### Applicable tag type

SEL

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bit item.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (unit: %)	0 to 100[%]	Input variable	REAL
CASIN_T	PV input (unit: %) (With tracking) (Indirect address)*1	0 to 100[%]	Input variable	DWORD
SELECT	Selection signal	TRUE: CASIN_T FALSE: PVP	Input variable	BOOL
MVN	Output to a module	NMIN to NMAX	Output variable	REAL
CASOUT	Cascade output (unit: %)	0 to 100[%]	Output variable	REAL

\*1 Connect CASOUT\_T of the primary loop with the input variable CASIN\_T.



## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
TRK	Tracking Flag	0: Not executed 1: Executed	0	User	INT
SVPTN_B4	CASIN_T Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL

## Tag data

For details on tag data that is read or written with this tag access FB, refer to the following.

☞ Page 866 SEL

## Processing details

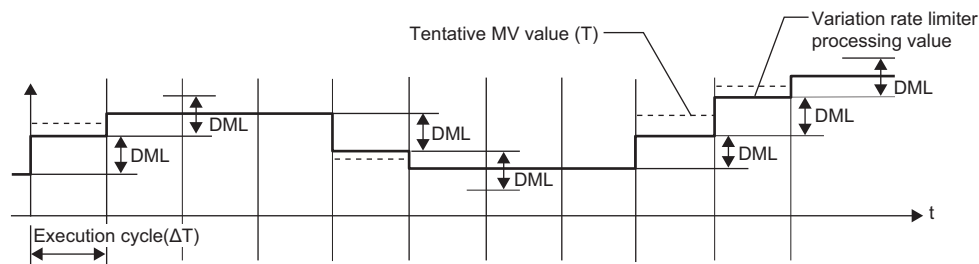
### Input selection processing

Selection signal (SELECT)	Result
FALSE	The input PVP (%) is selected.
TRUE	The input CASIN_T (%) is selected.

### Variation rate & high/low limiter

This function block checks the variation rate and high/low limits of the input value.

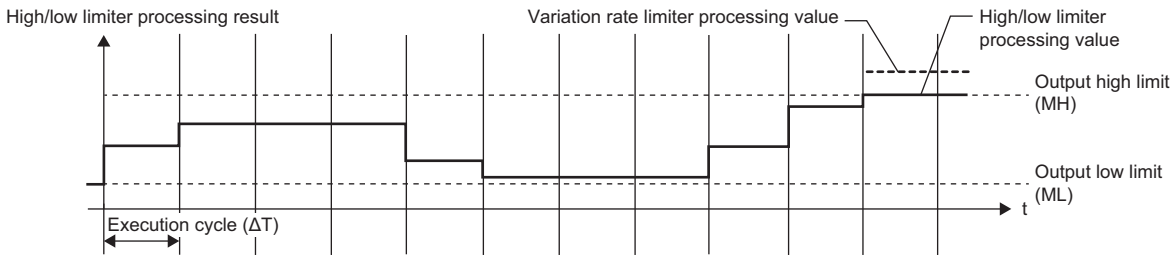
- Variation rate limiter



Condition	Variation rate limiter processing result	Alarm (ALM)
		Output variation rate limit (DMLA)
$ T - MV  \leq DML$	T	FALSE (Reset)
$T - MV > DML$	$MV + DML$	TRUE (Detected)
$T - MV < -DML$	$MV - DML$	TRUE (Detected)

T: Tentative manipulated value, MV: Manipulated value, DML: Output variation rate high limit value

• High/low limiter



Condition	High/low limiter processing result	Alarm (ALM)	
		Output Low Limit Alarm (MLA)	Output High Limit Alarm (MHA)
Variation rate limiter processing result > MH	MH	FALSE (Reset)	TRUE (Detected)
Variation rate limiter processing result < ML	ML	TRUE (Detected)	FALSE (Reset)
$ML \leq$ Variation rate limiter processing result $\leq$ MH	Variation rate limiter processing value	FALSE (Reset)	FALSE (Reset)

MH: Output high limit value, ML: Output low limit value

■Engineering value conversion

This function block converts an input value (%) to an engineering value.

$$PVn = \frac{RH-RL}{100} \times PVPn(\%) + RL$$

RH: Engineering value high limit

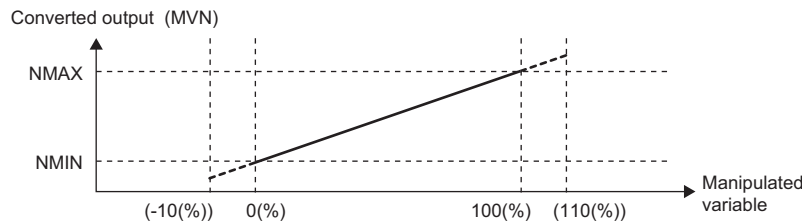
RL: Engineering value low limit

PVPn: PV input value (%)

PVn: PV1, PV2

■Output conversion

This function block performs output conversion processing.



$$\text{Converted output (MVN)} = \left\{ (NMAX - NMIN) \times \frac{MV}{100} \right\} + NMIN$$

NMAX: Output conversion high limit value

NMIN: Output conversion low limit

MV: Manipulated value (%)

MVN: Output conversion output value

■Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the variation rate & high/low limiter processing.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no DMLA, MHA, or MLA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERR1</li> <li>• DML1</li> <li>• MHI</li> <li>• MLI</li> </ul>
Disable alarm detection by loop stop processing	☞ Page 369 Loop stop processing

## ■ Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- Outputs (MVN) are held.
- The control mode is automatically switched to MANUAL.
- When DMLA, MHA, or MLA of the alarm (ALM) has occurred, the DMLA, MHA, or MLA is reset.
- No alarm is detected in the variation rate & high/low limiter processing.

## ■ Tracking processing

The following table shows whether tracking processing to the input variable CASIN\_T is performed or not.

Condition	Result
Tracking Flag (TRK)	
1	The input variable CASIN_T1 is tracked.
0	The input variable CASIN_T1 is not tracked.

## ■ Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation				
	Engineering value conversion	Variation rate & high/low limiter	Output conversion	Tracking	Alarm
MAN, CMV	○	×	○	○ <sup>*1</sup>	○ <sup>*3</sup>
AUT, CAS, CSV	○	○	○	○ <sup>*2</sup>	○ <sup>*3</sup>

\*1 Tracking is performed when the tracking flag (TRK) is 1.

\*2 Tracking is performed when the tracking flag (TRK) is 1 and DMLA, MHA, or MLA of the alarm (ALM) has occurred.

\*3 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

## Operation error

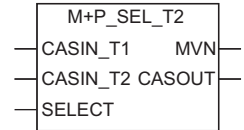
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data 1 (PVP), operation constant (NMAX, NMIN), tag data, or input data 2 (CASIN_T) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# 15.28 Loop Selector (Enable Tracking for primary loop by CASIN\_T1/\_T2) (M+P\_SEL\_T2)

## M+P\_SEL\_T2

This FB selects an input value depending on the setting of the select signal and outputs the result. The primary loop of the input 1 and 2 is tracked, but tracking from the secondary loop is not performed.

### FBD/LD



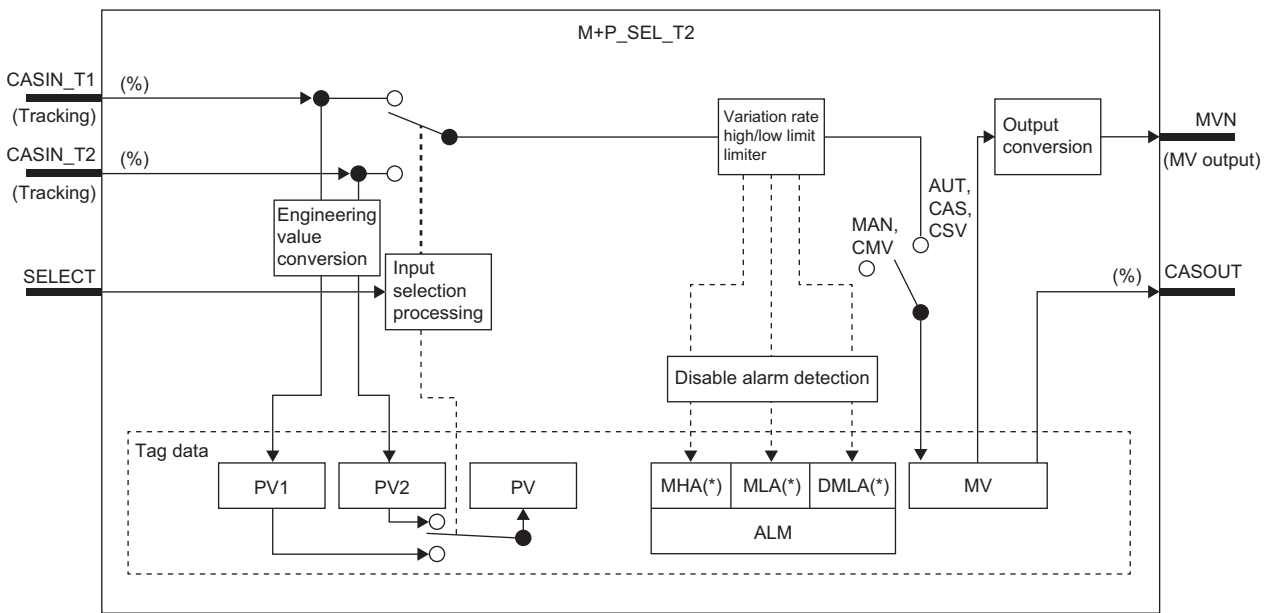
### Applicable tag type

SEL

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bit item.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
CASIN_T1	PV input (unit: %) (With tracking) (Indirect address)*1	0 to 100[%]	Input variable	DWORD
CASIN_T2	PV input (unit: %) (With tracking) (Indirect address)*1	0 to 100[%]	Input variable	DWORD
SELECT	Selection signal	TRUE: CASIN_T2 FALSE: CASIN_T1	Input variable	BOOL
MVN	Output to a module	NMIN to NMAX	Output variable	REAL
CASOUT	Cascade output (unit: %)	0 to 100[%]	Output variable	REAL

\*1 Connect CASOUT\_T of the primary loop with the input variable CASIN\_T1 and CASIN\_T2.

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
TRK	Tracking Flag	0: Not executed 1: Executed	0	User	INT
SVPTN_B1	CASIN_T1 Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SVPTN_B2	CASIN_T2 Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SVPTN_B3	CASIN_T1 Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
SVPTN_B4	CASIN_T2 Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL

## Tag data

For details on tag data that is read or written with this tag access FB, refer to the following.

Page 866 SEL

## Processing details

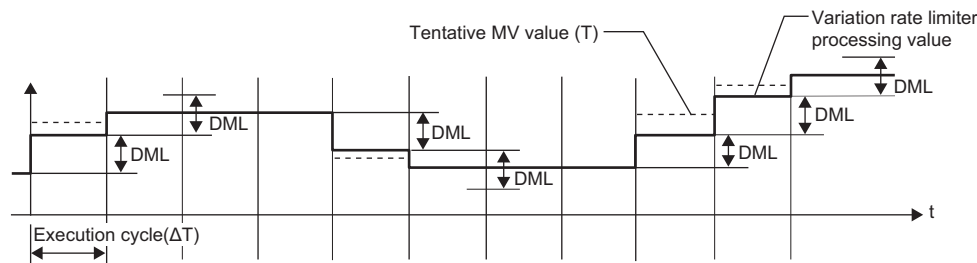
### Input selection processing

Selection signal (SELECT)	Result
FALSE	The input CASIN_T1 (%) is selected.
TRUE	The input CASIN_T2 (%) is selected.

### Variation rate & high/low limiter

This function block checks the variation rate and high/low limits of the input value.

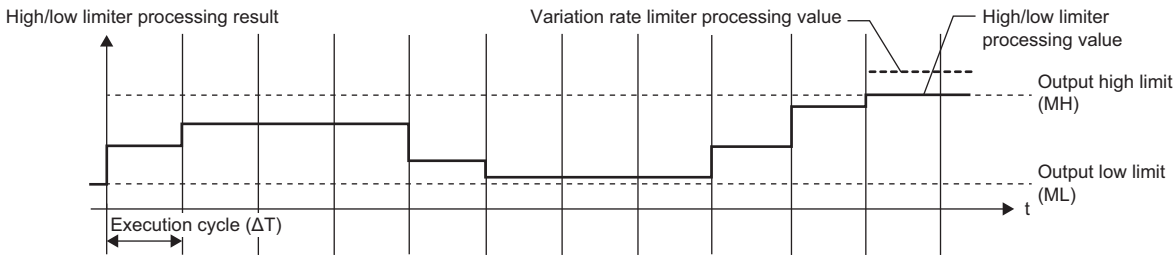
- Variation rate limiter



Condition	Variation rate limiter processing result	Alarm (ALM)
		Output variation rate limit (DMLA)
$ T - MV  \leq DML$	T	FALSE (Reset)
$T - MV > DML$	$MV + DML$	TRUE (Detected)
$T - MV < -DML$	$MV - DML$	TRUE (Detected)

T: Tentative manipulated value, MV: Manipulated value, DML: Output variation rate high limit value

• High/low limiter



Condition	High/low limiter processing result	Alarm (ALM)	
		Output Low Limit Alarm (MLA)	Output High Limit Alarm (MHA)
Variation rate limiter processing result > MH	MH	FALSE (Reset)	TRUE (Detected)
Variation rate limiter processing result < ML	ML	TRUE (Detected)	FALSE (Reset)
$ML \leq$ Variation rate limiter processing result $\leq$ MH	Variation rate limiter processing value	FALSE (Reset)	FALSE (Reset)

MH: Output high limit value, ML: Output low limit value

■Engineering value conversion

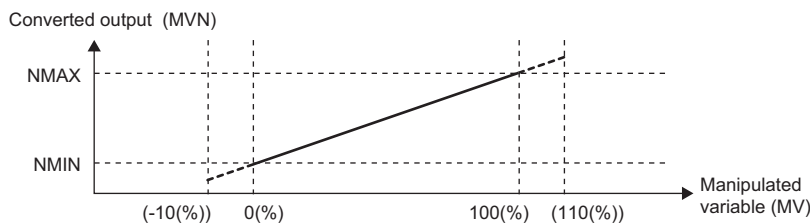
This function block converts an input value (%) to an engineering value.

$$PVn = \frac{RH-RL}{100} \times PVPn(\%) + RL$$

RH: Engineering value high limit  
 RL: Engineering value low limit  
 PVPn: PV input value (%)  
 PVn: PV1, PV2

■Output conversion

This function block performs output conversion processing.



$$\text{Converted output (MVN)} = \left\{ (NMAX - NMIN) \times \frac{MV}{100} \right\} + NMIN$$

NMAX: Output conversion high limit value  
 NMIN: Output conversion low limit  
 MV: Manipulated value (%)  
 MVN: Output conversion output value

■Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the variation rate & high/low limiter processing.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no DMLA, MHA, or MLA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DMLI</li> <li>• MHI</li> <li>• MLI</li> </ul>
Disable alarm detection by loop stop processing	☞ Page 373 Loop stop processing

### ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- Outputs (MVN) are held.
- The control mode is automatically switched to MANUAL.
- When DMLA, MHA, or MLA of the alarm (ALM) has occurred, the DMLA, MHA, or MLA is reset.
- No alarm is detected in the variation rate & high/low limiter processing.

### ■Tracking processing

The following table shows whether tracking processing to the input variable CASIN\_T is performed or not.

- Tracking of the input variable CASIN\_T1

Condition		Result
Tracking Flag (TRK)	CASIN_T1 Used (SVPTN_B1)	
1	FALSE	The input variable CASIN_T1 is tracked.
	TRUE	The input variable CASIN_T1 is not tracked.
0	FALSE or TRUE	

- Tracking of the input variable CASIN\_T2

Condition		Result
Tracking Flag (TRK)	CASIN_T2 Used (SVPTN_B2)	
1	FALSE	The input variable CASIN_T2 is tracked.
	TRUE	The input variable CASIN_T2 is not tracked.
0	FALSE or TRUE	

### ■Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation				
	Engineering value conversion	Variation rate & high/low limiter	Output conversion	Tracking	Alarm
MAN, CMV	○	×	○	○*1	○*3
AUT, CAS, CSV	○	○	○	○*2	○*3

\*1 Tracking is performed when the tracking flag (TRK) is 1.

\*2 Tracking is performed when the tracking flag (TRK) is 1 and DMLA, MHA, or MLA of the alarm (ALM) has occurred.

\*3 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

### Operation error

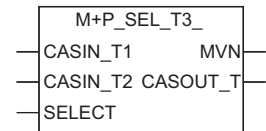
Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data 1 (CASIN_T1), operation constant (NMAX, NMIN), tag data, or input data 2 (CASIN_T2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# 15.29 Loop Selector (Enable Tracking from secondary loop to primary loop) (M+P\_SEL\_T3\_)

## M+P\_SEL\_T3\_

This FB selects two input values depending on the setting of the select signal and outputs the result. The primary loop of the input 1 and 2 is tracked from the secondary loop.

### FBD/LD



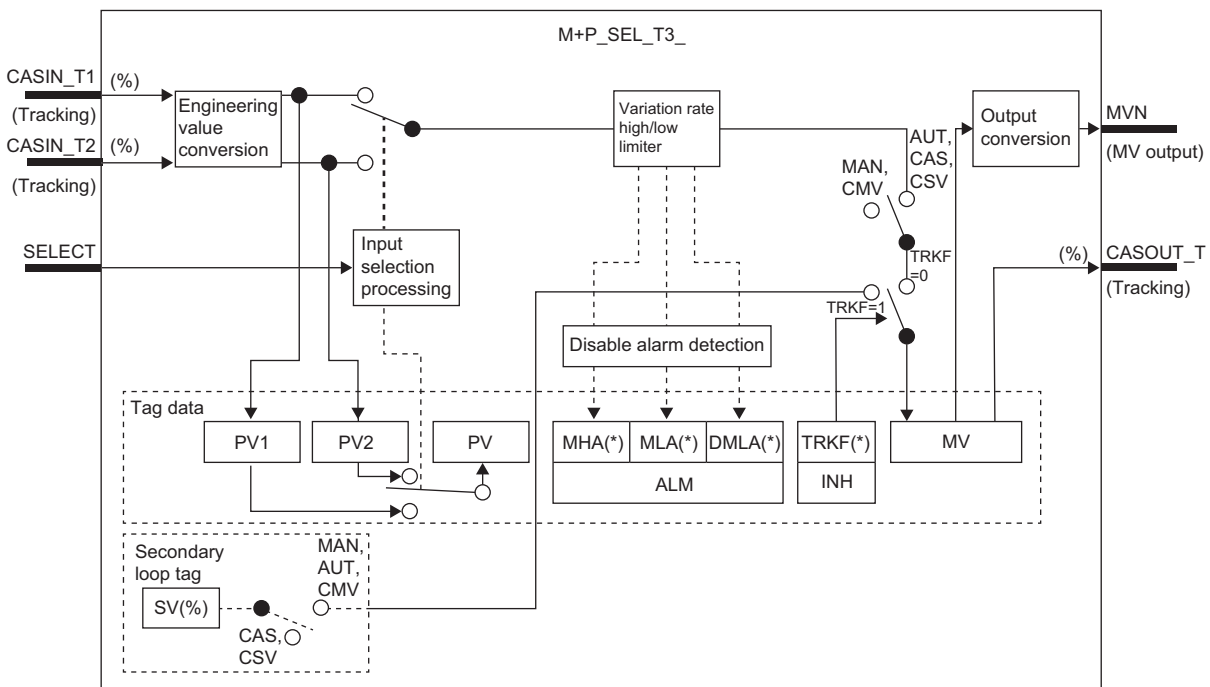
### Applicable tag type

SEL

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* Indicates bits item.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
CASIN_T1	PV input (unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
CASIN_T2	PV input (unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
SELECT	Selection signal	TRUE: CASIN_T2 FALSE: CASIN_T1	Input variable	BOOL
MVN	Output to a module	NMIN to NMAX	Output variable	REAL
CASOUT_T	Cascade output (unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD



## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
SVPTN_B1	CASIN_T1 Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SVPTN_B2	CASIN_T2 Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SVPTN_B3	CASIN_T1 Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
SVPTN_B4	CASIN_T2 Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
SVPTN_B5	Tracking to Non-selected loop	TRUE: Executed FALSE: Not executed	FALSE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T1 and CASIN\_T2.

## Tag data

For details on tag data that is read or written with this tag access FB, refer to the following.

☞ Page 866 SEL

## Processing details

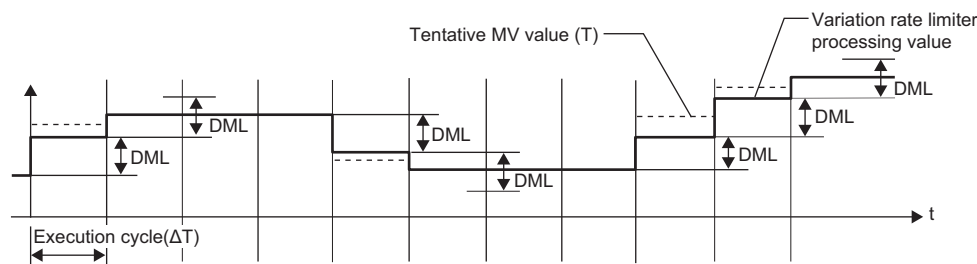
### Input selection processing

Selection signal (SELECT)	Result
FALSE	The input CASIN_T1 (%) is selected.
TRUE	The input CASIN_T2 (%) is selected.

### Variation rate & high/low limiter

This function block checks the variation rate and high/low limits of the input value.

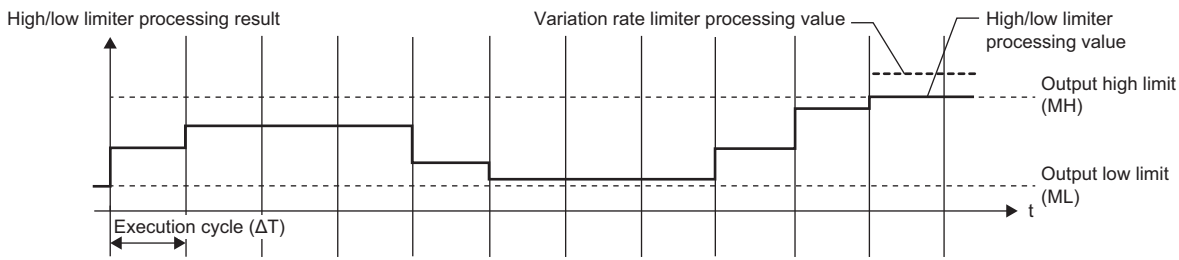
- Variation rate limiter



Condition	Variation rate limiter processing result	Alarm (ALM)
		Output variation rate limit (DMLA)
$ T - MV  \leq DML$	T	FALSE (Reset)
$T - MV > DML$	$MV + DML$	TRUE (Detected)
$T - MV < -DML$	$MV - DML$	TRUE (Detected)

T: Tentative manipulated value, MV: Manipulated value, DML: Output variation rate high limit value

• High/low limiter



Condition	High/low limiter processing result	Alarm (ALM)	
		Output Low Limit Alarm (MLA)	Output High Limit Alarm (MHA)
Variation rate limiter processing result > MH	MH	FALSE (Reset)	TRUE (Detected)
Variation rate limiter processing result < ML	ML	TRUE (Detected)	FALSE (Reset)
$ML \leq$ Variation rate limiter processing result $\leq$ MH	Variation rate limiter processing value	FALSE (Reset)	FALSE (Reset)

MH: Output high limit value, ML: Output low limit value

■Engineering value conversion

This function block converts an input value (%) to an engineering value.

$$PVn = \frac{RH-RL}{100} \times PVPn(\%) + RL$$

RH: Engineering value high limit

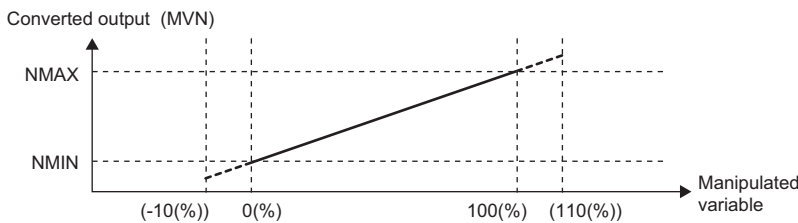
RL: Engineering value low limit

PVPn: PV input value (%)

PVn: PV1, PV2

■Output conversion

This function block performs output conversion processing.



$$\text{Converted output (MVN)} = \left\{ (NMAX - NMIN) \times \frac{MV}{100} \right\} + NMIN$$

NMAX: Output conversion high limit value

NMIN: Output conversion low limit

MV: Manipulated value (%)

MVN: Output conversion output value

■Disable alarm detection

This function block sets whether to detect alarms (ALM) or not in the variation rate & high/low limiter processing.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no DMLA, MHA, or MLA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DMLI</li> <li>• MHI</li> <li>• MLI</li> </ul>
Disable alarm detection by loop stop processing	☞ Page 377 Loop stop processing

### Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) is TRUE, this function block performs the following processing.

- Outputs (MVN) are held.
- The control mode is automatically switched to MANUAL.
- When DMLA, MHA, or MLA of the alarm (ALM) has occurred, the DMLA, MHA, or MLA is reset.
- No alarm is detected in the variation rate & high/low limiter processing.

### Tracking processing

The following table shows whether tracking processing to input variables CASIN\_T1 and CASIN\_T2 are performed or not.

- Tracking of the input variable CASIN\_T1

Condition				Result
Tracking Flag (TRK)	CASIN_T1 Used (SVPTN_B1)	SELECT	SVPTN_B5	
1	FALSE	FALSE	TRUE or FALSE	The input variable CASIN_T1 is tracked.
		TRUE	TRUE	The input variable CASIN_T1 is not tracked.
	TRUE	TRUE or FALSE	FALSE	
0	TRUE or FALSE			

- Tracking of the input variable CASIN\_T2

Condition				Result
Tracking Flag (TRK)	CASIN_T2 Used (SVPTN_B2)	SELECT	SVPTN_B5	
1	FALSE	FALSE	TRUE	The input variable CASIN_T2 is tracked.
			FALSE	The input variable CASIN_T2 is not tracked.
	TRUE	TRUE or FALSE	TRUE or FALSE	The input variable CASIN_T2 is tracked.
0	TRUE or FALSE	TRUE or FALSE		The input variable CASIN_T2 is not tracked.

Set the same program execution cycle and the control cycle (CT) for both of the primary loop and the secondary loop.

### Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation				
	Engineering value conversion	Variation rate & high/low limiter	Output conversion	Tracking	Alarm
MAN, CMV	○	×	○	○*1	○*3
AUT, CAS, CSV	○	○	○	○*2	○*4

\*1 Tracking to a selected loop and a non-selected loop is performed when the tracking flag (TRK) is 1.

\*2 Tracking to a non-selected loop is performed when the tracking flag (TRK) is 1.

Tracking to a selected loop is performed when DMLA, MHA, or MLA of the alarm (ALM) has occurred.

\*3 An alarm (ALM) whose corresponding bit is TRUE (detected) is reset, and the alarm will not be detected.

\*4 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

### Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (CASIN_T1, CASIN_T2, MV), tag data, or operation constant (RH, RL, NMAX, NMIN) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# 16 TAG SPECIAL

The following FB changes the control mode.



To set initial values of public variables of the tag access FB arranged on a user-defined tag FB in "FB Property" of the engineering tool, refer to the following.

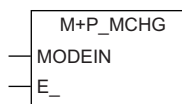
GX Works3 Operating Manual

## 16.1 Change Control Mode (M+P\_MCHG)

### M+P\_MCHG

This function block switches the control mode to one of MANUAL, AUTO, CASCADE, COMPUTER MV, COMPUTER SV, CASCADE DIRECT according to a mode change signal.

#### FBD/LD



#### Applicable tag type

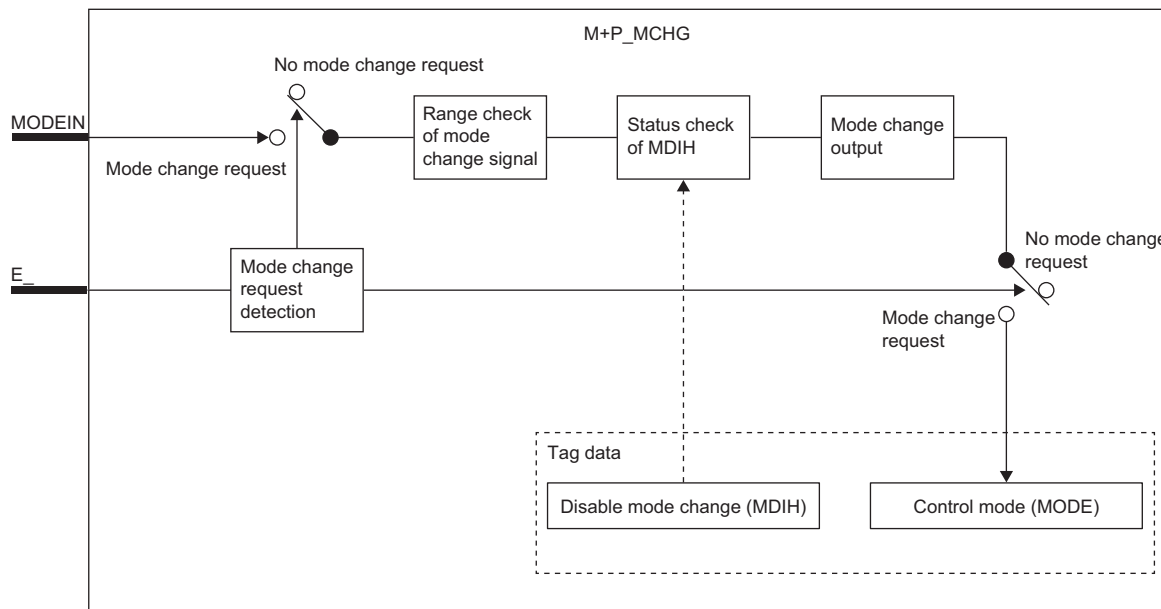
PID, 2PID, 2PIDH, PIDP, SPI, IPD, BPI, R, ONF2, ONF3, PGS, PGS2, MOU, SWM, MWM, SEL, PVAL, HTCL, NREV, REV, MVAL1, MVAL2, PB

#### Control mode

MAN	AUT	CAS*1	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

\*1 Transition to CASDR is possible.

#### Block diagram



## Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
MODEIN	Mode Change Signal	1 to 6 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV 6: CASDR	Input variable	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	Input variable	BOOL


### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
BNAL_MODE <sup>*1</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT

\*1 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

### Tag data

For details on tag data that is read or written with this tag access FB, refer to the following.

 Page 812 Tag Data List

## Processing details

### Mode change signal range check

This function block checks the mode change signal range. Set a value within the range of 1 to 6 for the mode change signal.

MODEIN (mode change signal)	6	5	4	3	2	1
MODE (control mode)	CASDR	CSV	CMV	CAS	AUT	MAN

CASDR: CASCADE DIRECT

CSV: COMPUTER SV

CMV: COMPUTER MV

CAS: CASCADE

AUT: AUTO

MAN: MANUAL

- Change CASDR and CASI into FALSE to switch to the CASCADE DIRECT mode.
- This function block turns on the bits of CAS and CASDR of the control mode (MODE) of the tag data at the mode transition to the CASCADE DIRECT mode.
- If a value outside the range of 1 to 6 is set in MODEIN, the mode selection transition disable check and the mode selection output processing will not be performed. When the control mode of tag data is MODE, the last value is held.

### Mode switching disable check

Switching of the control mode whose bit of the mode disable (MDIH) of tag data is TRUE (Valid) will be disabled. (The mode switching output processing is not performed.)

## ■ Switching request and mode switching output

- When a sensor error (SEA) occurs or an output open (OOA) occurs, this function block sets whether or not to switch the control mode to MANUAL according to BNAL\_MODE. However, no processing is performed when the Disable MANUAL bit (MANI) is ON.\*1

Condition	Processing result
BNAL_MODE(0)	Comply with the setting common to all tags.*2 <ul style="list-style-type: none"> <li>"Yes" is selected for "Switch the Control Mode to MANUAL"; Switch to MANUAL mode.</li> <li>"No" is selected for "Switch the Control Mode to MANUAL"; Do not switch to MANUAL mode.</li> </ul>
BNAL_MODE(1)	Set the individual operation for each tag. Switch to MANUAL mode.
BNAL_MODE(2)	Set the individual operation for each tag. Do not switch to MANUAL mode.

\*1 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*2 The setting common to all tags is set in the process control extended setting in the engineering tool.

🔗 [Options] ⇨ [Convert] ⇨ [Process Control Extension Setting] ⇨ [I/O Control] ⇨ [Disconnection Detection]

- When the enable mode change (E\_) is TRUE, this function block changes the mode according to the mode change signal and sets the corresponding bit of the control mode (MODE) of tag data to TRUE. (🔗 Page 812 Tag Data List)

Condition					Control mode (MODE) of tag data
Mode change request (E_)	Mode change signal (MODEIN)	Mode change signal range check	Mode switching disable check	Mode switching output	
FALSE	1 to 6	Valid	Stop	Stop	The last value is held.
	Other than 1 to 6	Invalid	Stop	Stop	The last value is held.
TRUE	1 to 6	Valid	Execute	Execute	The corresponding bit is set to TRUE.
	Other than 1 to 6	Invalid	Stop	Stop	The last value is held.

- The control mode is switched to MANUAL according to the mode switching processing from the secondary loop in the cascade connection. (🔗 Page 381 M+P\_MCHGPRMRY)

## ■ Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation			
	Mode change signal range check	Mode transition disable mode	Change request detection	Mode switching output
MAN, CMV, AUT, CAS, CSV, CASDR	○	○	○	○

## Operation error

There is no operation error.

# 16.2 Primary Loop Control Mode Switching (M+P\_MCHGPRMRY)

## M+P\_MCHGPRMRY

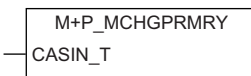
• Can be used if the version of the function block is 1.070Y or later.

For the cascade connection, this function block switches the control mode of the primary loop to MANUAL when an alarm occurs or when the control mode is switched.



When using this tag access FB with the user-defined tag FB, arrange it so that the execution order is after the output processing.

### FBD/LD



### Applicable tag type

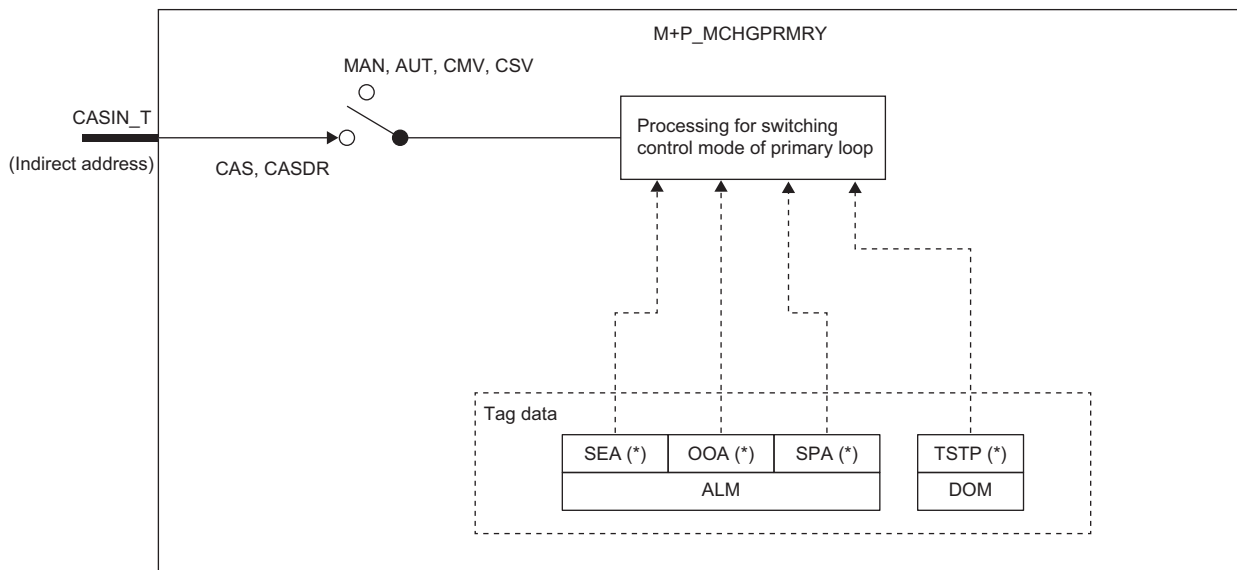
PID, 2PID, 2PIDH, PIDP, SPI, IPD, BPI, R, ONF2, ONF3, SEL, PVAL, HTCL

### Control mode

MAN	AUT	CAS* <sup>1</sup>	CMV	CSV
×	×	○	×	×

\*1 Transition to CASDR is possible.

### Block diagram



\* Indicates bits item.

## Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
CASIN_T	Cascade input (Indirect address)	—	Input variable	DWORD

### ■Public variable (operation constant)

Variable name	Description *1	Recommended range	Initial value	Set by	Data type
SEA_EN	When a sensor error occurs, switches the primary loop to MANUAL mode	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
OOA_EN	When output open occurs, switches the primary loop to MANUAL mode				
SPA_EN	When a stop alarm occurs, switches the primary loop to MANUAL mode				
TSTP_EN	When tag stop occurs, switches the primary loop to MANUAL mode				
CASCASDR_EN	When the control mode is changed (CAS/CASDR→AUT/MAN/CSV/CMV), switches the primary loop to MANUAL mode				

\*1 When TRUE (switch) is set, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

### ■Tag data

For details on tag data that is read or written with this tag access FB, refer to the following.

☞ Page 812 Tag Data List

## Processing details

### ■Primary loop mode switching processing

When any of the following conditions is met, the processing to switch the primary loop control mode to MANUAL is performed.

- When SEA\_EN is TRUE and sensor error (SEA) of the alarm (ALM) occurs \*1
- When OOA\_EN is TRUE and output open (OOA) of the alarm (ALM) is TRUE
- When SPA\_EN is TRUE and stop alarm (SPA) of the alarm (ALM) is TRUE
- When TSTP\_EN is TRUE and tag stop (TSTP) of the monitor output buffer (DOM) is TRUE \*2
- When CASCASDR\_EN is TRUE and the mode changed from CAS/CASDR to MAN/CMV/AUT/CSV

\*1 When sensor error inhibit (SEI) of disable alarm detection (INH) or disable all alarms (ERRI) is TRUE, mode switching processing is not performed.

\*2 When tag stop inhibit (TSTPI) of mode disable (MDIH) is TRUE, mode switching processing is not performed.

### ■Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation
	Primary loop mode switching processing
MAN, CMV, AUT, CSV	×
CAS, CASDR	○

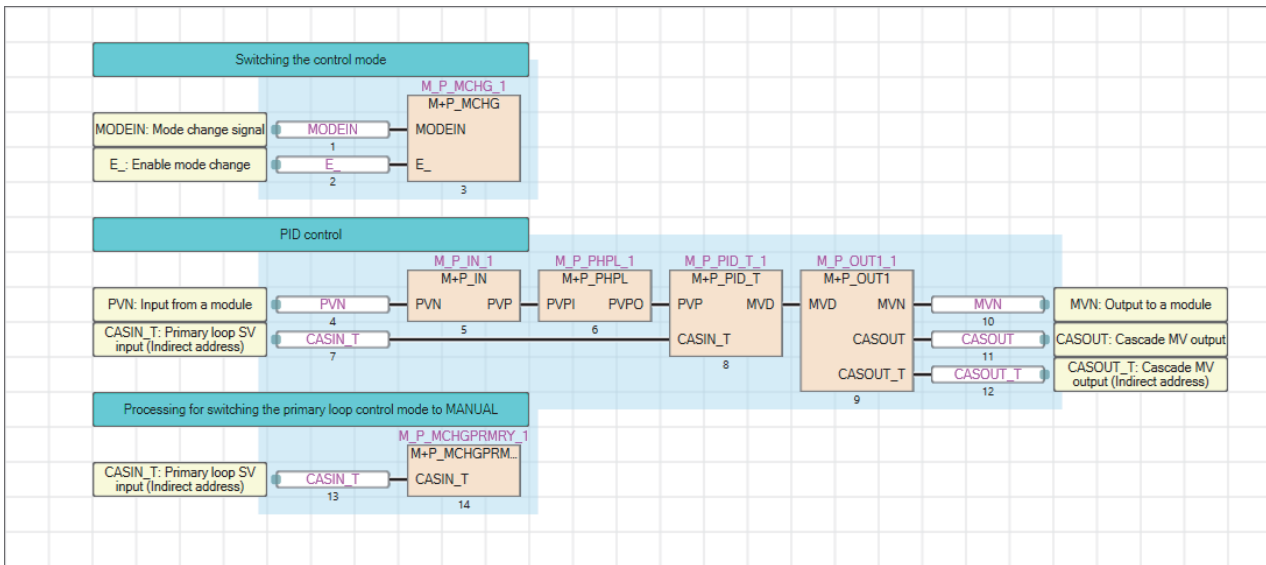
## Operation error

There is no operation error.



## Program example

When used with FB for velocity type PID control



### Point

When using this tag access FB with the user-defined tag FB, arrange it so that the execution order is after the output processing.



This part consists of the following chapters.

17 LOOP TAG

---

18 STATUS TAG

---

19 ALARM TAG

---

20 MESSAGE TAG

---

# 17 LOOP TAG

The following FBs perform loop control processing including ratio control, various PID controls, two-position (on/off) control, three-position (on/off) control, program setter, and loop selector.

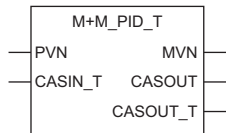
## 17.1 Velocity Type PID Control (Enable Tracking for primary loop) (M+M\_PID\_T)

### M+M\_PID\_T

This FB performs velocity type PID control combining the functions of M+P\_IN, M+P\_PHPL, M+P\_PID\_T, and M+P\_OUT1 into a single FB.

The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.

#### FBD/LD



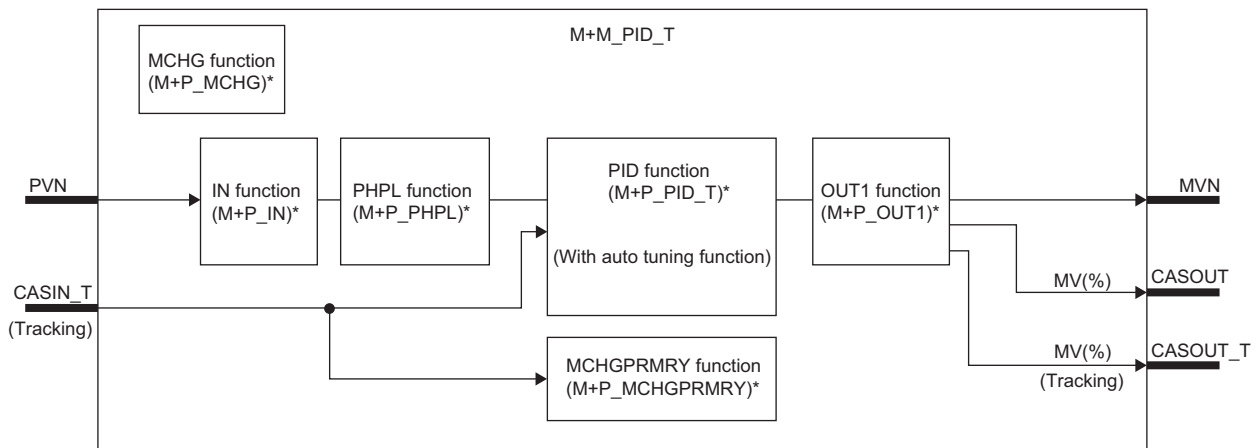
#### Applicable tag type

PID

#### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

#### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### ■ Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN_T	Primary loop SV input (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVN	Output to a module	OUT1_NMIN to OUT1_NMAX	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

## ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*5*6</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*2*3</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
PID_MTD	Derivative Gain	0 to 9999	8.0	User	REAL
PID_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PID_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
PID_TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
PID_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
PID_SVPTN_B1	Setting Value (SV) Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
OUT1_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
OUT1_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
OUT1_SEA_OTYPE <sup>*5*6</sup>	Hold processing selection for output function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT
MCHG_BNAL_MODE <sup>*2*3</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT
MCHGPRMRY_SEA_EN <sup>*2</sup>	When a sensor error occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_OOA_EN <sup>*2</sup>	When output open occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_SPA_EN <sup>*2</sup>	When a stop alarm occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_CASCASDR_EN <sup>*2</sup>	When the control mode is changed (CAS→AUT/MAN/CSV/CMV), switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*2 This public variable was added in version 1.070Y of the function block.

\*3 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*4 When TRUE (switch) is set, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*5 This public variable was added in version 1.085P of the function block.

\*6 If the version of the function block is lower than 1.085P, complies with the setting common to all tags.

## ■ Public variable (others) \*1

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■ Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

☞ Page 813 PID

### Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
PID function	M+P_PID_T	Page 230 M+P_PID_T
OUT1 function	M+P_OUT1	Page 189 M+P_OUT1
MCHG function	M+P_MCHG	Page 378 M+P_MCHG
MCHGPRMRY function*1	M+P_MCHGPRMRY	Page 381 M+P_MCHGPRMRY

\*1 This function was added in version 1.070Y of the function block.

### Operation error

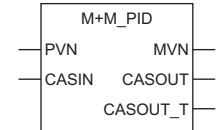
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.2 Velocity Type PID Control (Disable Tracking for primary loop) (M+M\_PID)

## M+M\_PID

This FB performs velocity type PID control combining the functions of M+P\_IN, M+P\_PHPL, M+P\_PID, and M+P\_OUT1 into a single FB.

### FBD/LD



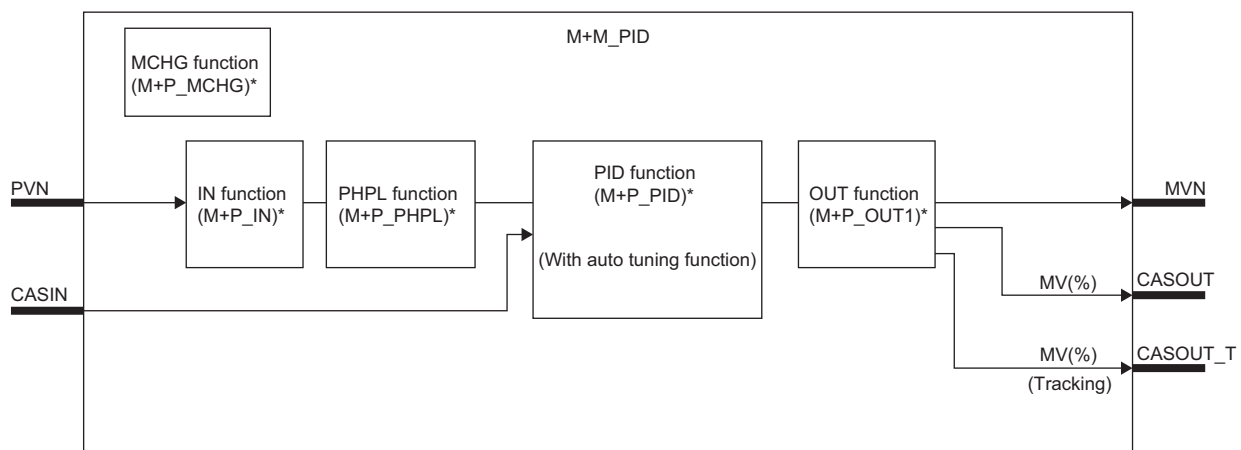
### Applicable tag type

PID

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN	Primary loop SV input (Unit: %)	0 to 100[%]	Input variable	REAL
MVN	Output to a module	OUT1_NMIN to OUT1_NMAX	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD



## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*2</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*1</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
PID_MTD	Derivative Gain	0 to 9999	8.0	User	REAL
PID_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PID_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
PID_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
OUT1_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
OUT1_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
OUT1_SEA_OTYPE <sup>*2</sup>	Hold processing selection for output function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT
MCHG_BNAL_MODE <sup>*1</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT

\*1 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

\*2 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

## Public variable (others) <sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

Page 813 PID

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
PID function	M+P_PID	Page 236 M+P_PID
OUT1 function	M+P_OUT1	Page 189 M+P_OUT1
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

## Operation error

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

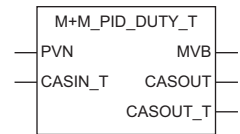
# 17.3 Velocity Type PID Control and DUTY Output (Enable Tracking for primary loop) (M+M\_PID\_DUTY\_T)

## M+M\_PID\_DUTY\_T

This FB performs velocity type PID control and duty output combining the functions of M+P\_IN, M+P\_PHPL, M+P\_PID\_T, and M+P\_DUTY into a single FB.

The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.

### FBD/LD



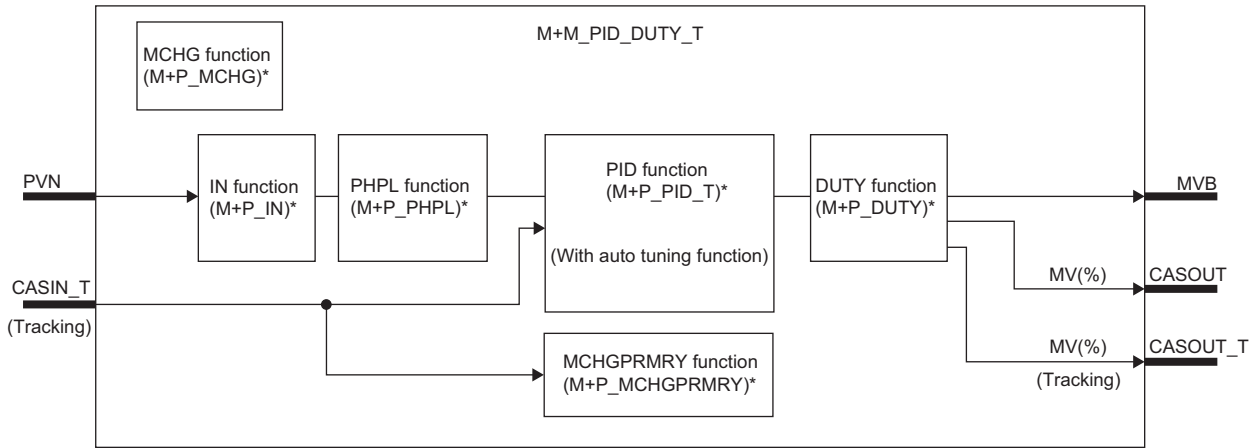
### Applicable tag type

PID

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN_T	Primary loop SV input (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVB	Bit output to a module	TRUE, FALSE	Output variable	BOOL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*5*6</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*2*3</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
PID_MTD	Derivative Gain	0 to 9999	8.0	User	REAL
PID_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PID_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
PID_TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
PID_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
PID_SVPTN_B1	Setting Value (SV) Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
DUTY_SEA_OTYPE <sup>*5*6</sup>	Hold processing selection for output function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT
MCHG_BNAL_MODE <sup>*2*3</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT
MCHGPRMRY_SEA_EN <sup>*2</sup>	When a sensor error occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_OOA_EN <sup>*2</sup>	When output open occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_SPA_EN <sup>*2</sup>	When a stop alarm occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_CASCASDR_EN <sup>*2</sup>	When the control mode is changed (CAS→AUT/MAN/CSV/CMV), switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*2 This public variable was added in version 1.070Y of the function block.

\*3 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*4 When TRUE (switch) is set, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*5 This public variable was added in version 1.085P of the function block.

\*6 If the version of the function block is lower than 1.085P, complies with the setting common to all tags.

## ■ Public variable (others)<sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■ Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

📖 Page 813 PID

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
PID function	M+P_PID_T	Page 230 M+P_PID_T
DUTY function	M+P_DUTY	Page 208 M+P_DUTY
MCHG function	M+P_MCHG	Page 378 M+P_MCHG
MCHGPRMRY function <sup>*1</sup>	M+P_MCHGPRMRY	Page 381 M+P_MCHGPRMRY

\*1 This function was added in version 1.070Y of the function block.

## Operation error

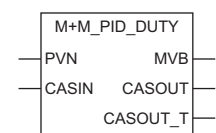
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.4 Velocity Type PID Control and DUTY Output (Disable Tracking for primary loop) (M+M\_PID\_DUTY)

## M+M\_PID\_DUTY

This FB performs velocity type PID control and duty output combining the functions of M+P\_IN, M+P\_PHPL, M+P\_PID, and M+P\_DUTY into a single FB.

### FBD/LD



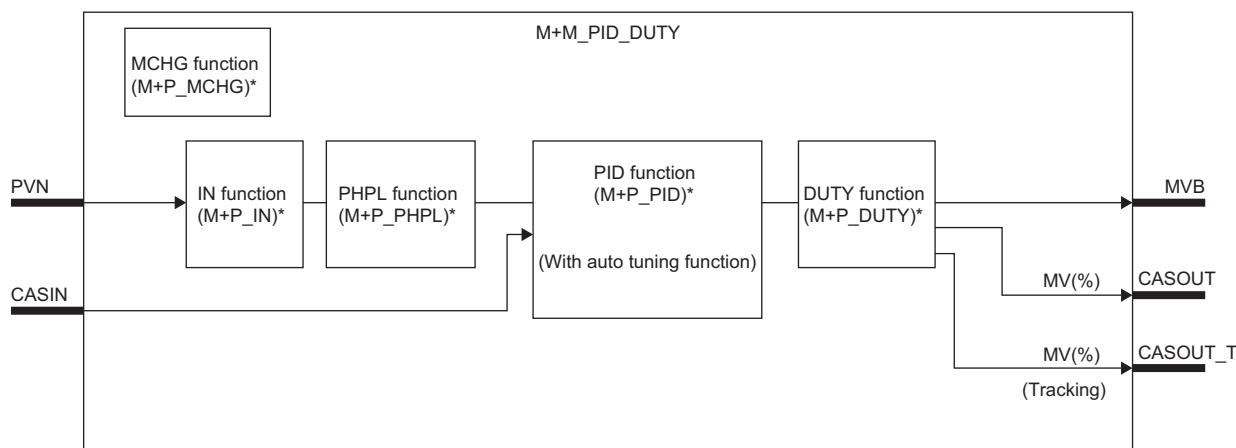
### Applicable tag type

PID

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN	Primary loop SV input (Unit: %)	0 to 100[%]	Input variable	REAL
MVB	Bit output to a module	TRUE, FALSE	Output variable	BOOL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

## ■ Public variable (operation constant)

- Operation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*2</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*1</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
PID_MTD	Derivative Gain	0 to 9999	8.0	User	REAL
PID_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PID_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
PID_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
DUTY_SEA_OTYPE <sup>*2</sup>	Hold processing selection for output function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT
MCHG_BNAL_MODE <sup>*1</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT

\*1 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

\*2 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

## ■ Public variable (others) <sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■ Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

📖 Page 813 PID

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
PID function	M+P_PID	Page 236 M+P_PID
DUTY function	M+P_DUTY	Page 208 M+P_DUTY
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

## Operation error

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.



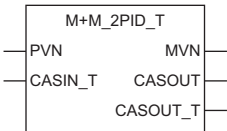
# 17.5 2-degree-of-freedom PID Control (Enable Tracking for primary loop) (M+M\_2PID\_T)

## M+M\_2PID\_T

This FB performs two-degree-of-freedom PID control combining the functions of M+P\_IN, M+P\_PHPL, M+P\_2PID\_T, and M+P\_OUT1 into a single FB.

The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.

### FBD/LD



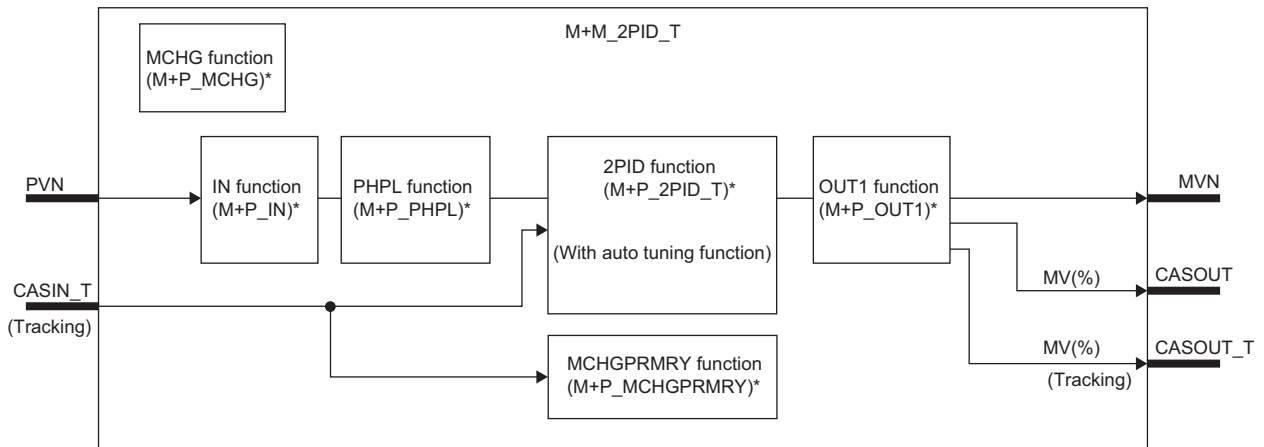
### Applicable tag type

2PID

### Control mode

MAN	AUT	CAS	CMV	CSV
○	○	○	○	○

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN_T	Primary loop SV input (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVN	Output to a module	OUT1_NMIN to OUT1_NMAX	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*5*6</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*2*3</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
PID2_MTD	Derivative Gain	0 to 9999	8.0	User	REAL
PID2_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PID2_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
PID2_TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
PID2_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
PID2_SVPTN_B1	Setting Value (SV) Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
OUT1_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
OUT1_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
OUT1_SEA_OTYPE <sup>*5*6</sup>	Hold processing selection for output function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT
MCHG_BNAL_MODE <sup>*2*3</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT
MCHGPRMRY_SEA_EN <sup>*2</sup>	When a sensor error occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_OOA_EN <sup>*2</sup>	When output open occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_SPA_EN <sup>*2</sup>	When a stop alarm occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_CASCASDR_EN <sup>*2</sup>	When the control mode is changed (CAS→AUT/MAN/CSV/CMV), switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*2 This public variable was added in version 1.070Y of the function block.

\*3 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*4 When TRUE (switch) is set, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*5 This public variable was added in version 1.085P of the function block.

\*6 If the version of the function block is lower than 1.085P, complies with the setting common to all tags.

## ■ Public variable (others)<sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL


- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■ Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

 Page 818 2PID

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
2PID function	M+P_2PID_T	Page 242 M+P_2PID_T
OUT1 function	M+P_OUT1	Page 189 M+P_OUT1
MCHG function	M+P_MCHG	Page 378 M+P_MCHG
MCHGPRMRY function <sup>*1</sup>	M+P_MCHGPRMRY	Page 381 M+P_MCHGPRMRY

\*1 This function was added in version 1.070Y of the function block.

## Operation error

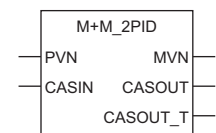
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.6 2-degree-of-freedom PID Control (Disable Tracking for primary loop) (M+M\_2PID)

## M+M\_2PID

This FB performs two-degree-of-freedom PID control combining the functions of M+P\_IN, M+P\_PHPL, M+P\_2PID, and M+P\_OUT1 into a single FB.

### FBD/LD



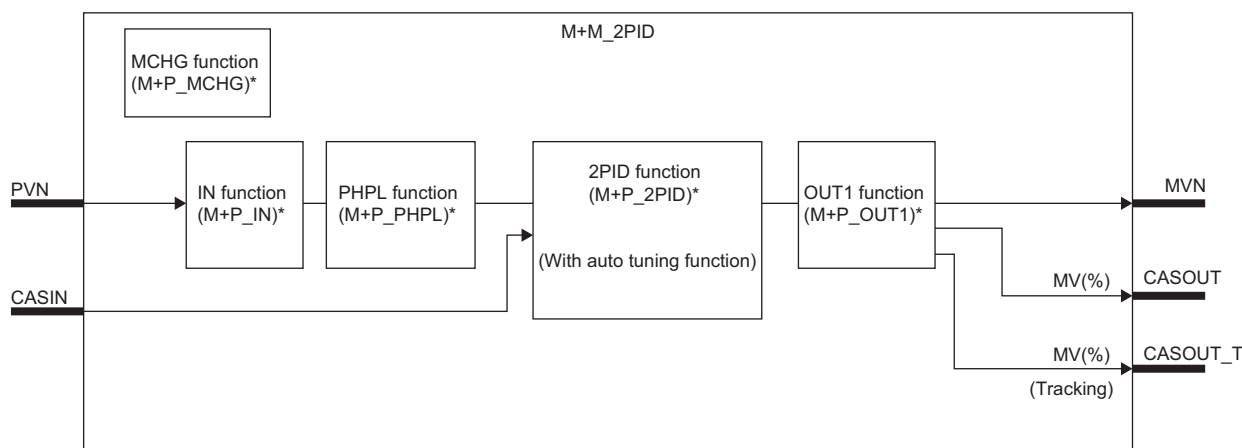
### Applicable tag type

2PID

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN	Primary loop SV input (Unit: %)	0 to 100[%]	Input variable	REAL
MVN	Output to a module	OUT1_NMIN to OUT1_NMAX	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*2</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*1</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
PID2_MTD	Derivative Gain	0 to 9999	8.0	User	REAL
PID2_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PID2_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
PID2_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
OUT1_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
OUT1_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
OUT1_SEA_OTYPE <sup>*2</sup>	Hold processing selection for output function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT
MCHG_BNAL_MODE <sup>*1</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT

\*1 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

\*2 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

## Public variable (others) <sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

☞ Page 818 2PID

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
2PID function	M+P_2PID	Page 248 M+P_2PID
OUT1 function	M+P_OUT1	Page 189 M+P_OUT1
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

## Operation error

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

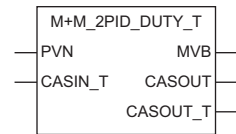
# 17.7 2-degree-of-freedom PID Control and DUTY O/P (Enable Tracking for primary loop) (M+M\_2PID\_DUTY\_T)

## M+M\_2PID\_DUTY\_T

This FB performs two-degree-of-freedom PID control and duty output combining the functions of M+P\_IN, M+P\_PHPL, M+P\_2PID\_T, and M+P\_DUTY into a single FB.

The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.

### FBD/LD



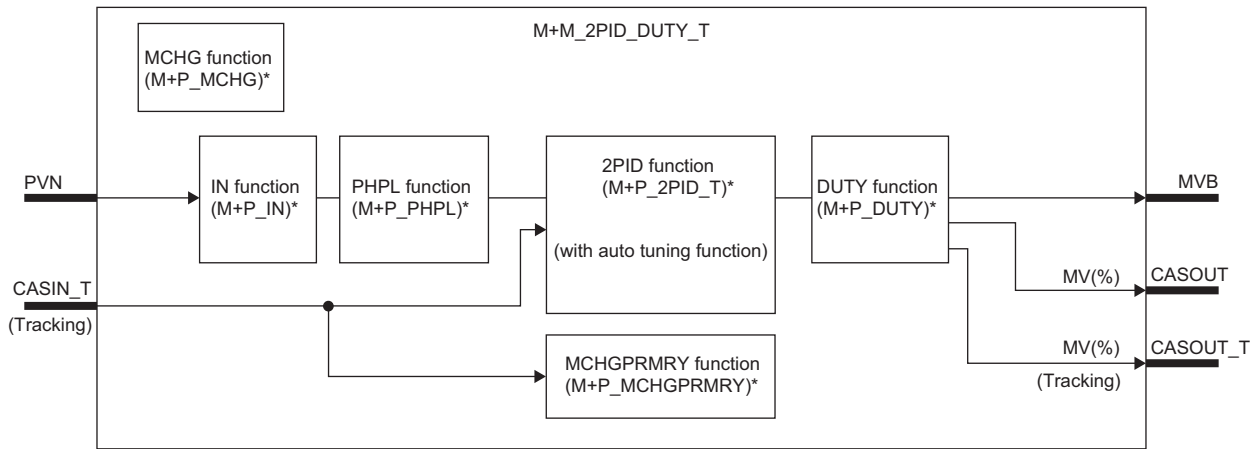
### Applicable tag type

2PID

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN_T	Primary loop SV input (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVB	Bit output to a module	TRUE, FALSE	Output variable	BOOL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*5*6</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*2*3</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
PID2_MTD	Derivative Gain	0 to 9999	8.0	User	REAL
PID2_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PID2_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
PID2_TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
PID2_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
PID2_SVPTN_B1	Setting Value (SV) Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
DUTY_SEA_OTYPE <sup>*5*6</sup>	Hold processing selection for output function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT
MCHG_BNAL_MODE <sup>*2*3</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT
MCHGPRMRY_SEA_EN <sup>*2</sup>	When a sensor error occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_OOA_EN <sup>*2</sup>	When output open occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_SPA_EN <sup>*2</sup>	When a stop alarm occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_CASCASDR_EN <sup>*2</sup>	When the control mode is changed (CAS→AUT/MAN/CSV/CMV), switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*2 This public variable was added in version 1.070Y of the function block.

\*3 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*4 When TRUE (switch) is set, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*5 This public variable was added in version 1.085P of the function block.

\*6 If the version of the function block is lower than 1.085P, complies with the setting common to all tags.



## ■ Public variable (others)<sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■ Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

📖 Page 818 2PID

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
2PID function	M+P_2PID_T	Page 242 M+P_2PID_T
DUTY function	M+P_DUTY	Page 208 M+P_DUTY
MCHG function	M+P_MCHG	Page 378 M+P_MCHG
MCHGPRMRY function <sup>*1</sup>	M+P_MCHGPRMRY	Page 381 M+P_MCHGPRMRY

\*1 This function was added in version 1.070Y of the function block.

## Operation error

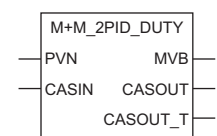
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.8 2-degree-of-freedom PID Control and DUTY O/P (Disable Tracking for primary loop) (M+M\_2PID\_DUTY)

## M+M\_2PID\_DUTY

This FB performs two-degree-of-freedom PID control and duty output combining the functions of M+P\_IN, M+P\_PHPL, M+P\_2PID, and M+P\_DUTY into a single FB.

### FBD/LD



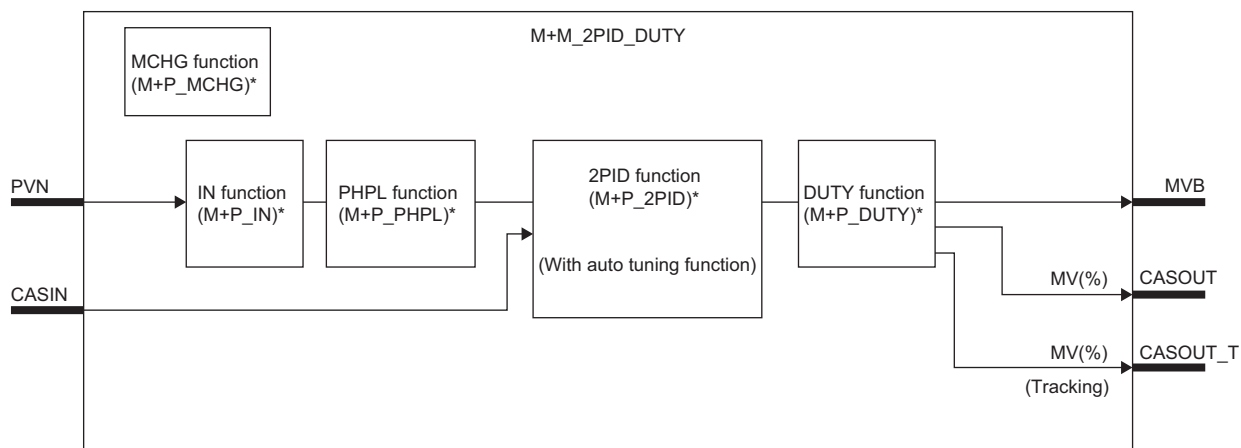
### Applicable tag type

2PID

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN	Primary loop SV input (Unit: %)	0 to 100[%]	Input variable	REAL
MVB	Bit output to a module	TRUE, FALSE	Output variable	BOOL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*2</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*1</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
PID2_MTD	Derivative Gain	0 to 9999	8.0	User	REAL
PID2_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PID2_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
PID2_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
DUTY_SEA_OTYPE <sup>*2</sup>	Hold processing selection for output function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT
MCHG_BNAL_MODE <sup>*1</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT

\*1 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

\*2 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

## Public variable (others) <sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

📖 Page 818 2PID

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
2PID function	M+P_2PID	Page 248 M+P_2PID
DUTY function	M+P_DUTY	Page 208 M+P_DUTY
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

## Operation error

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

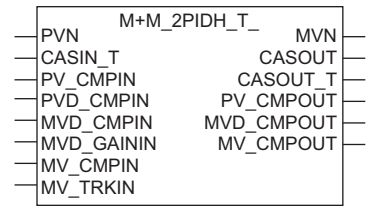
# 17.9 2-degree-of-freedom Advanced PID Control (Enable Tracking for primary loop) (M+M\_2PIDH\_T\_)

## M+M\_2PIDH\_T\_

This FB performs two-degree-of-freedom PID control combining the functions of M+P\_IN, M+P\_PHPL, M+P\_2PIDH\_T\_, and M+P\_OUT3\_ into a single FB with PV/MV correction.

The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, tag stop, or control mode change occurs, the primary loop control mode is switched to MANUAL.

### FBD/LD



### Applicable tag type

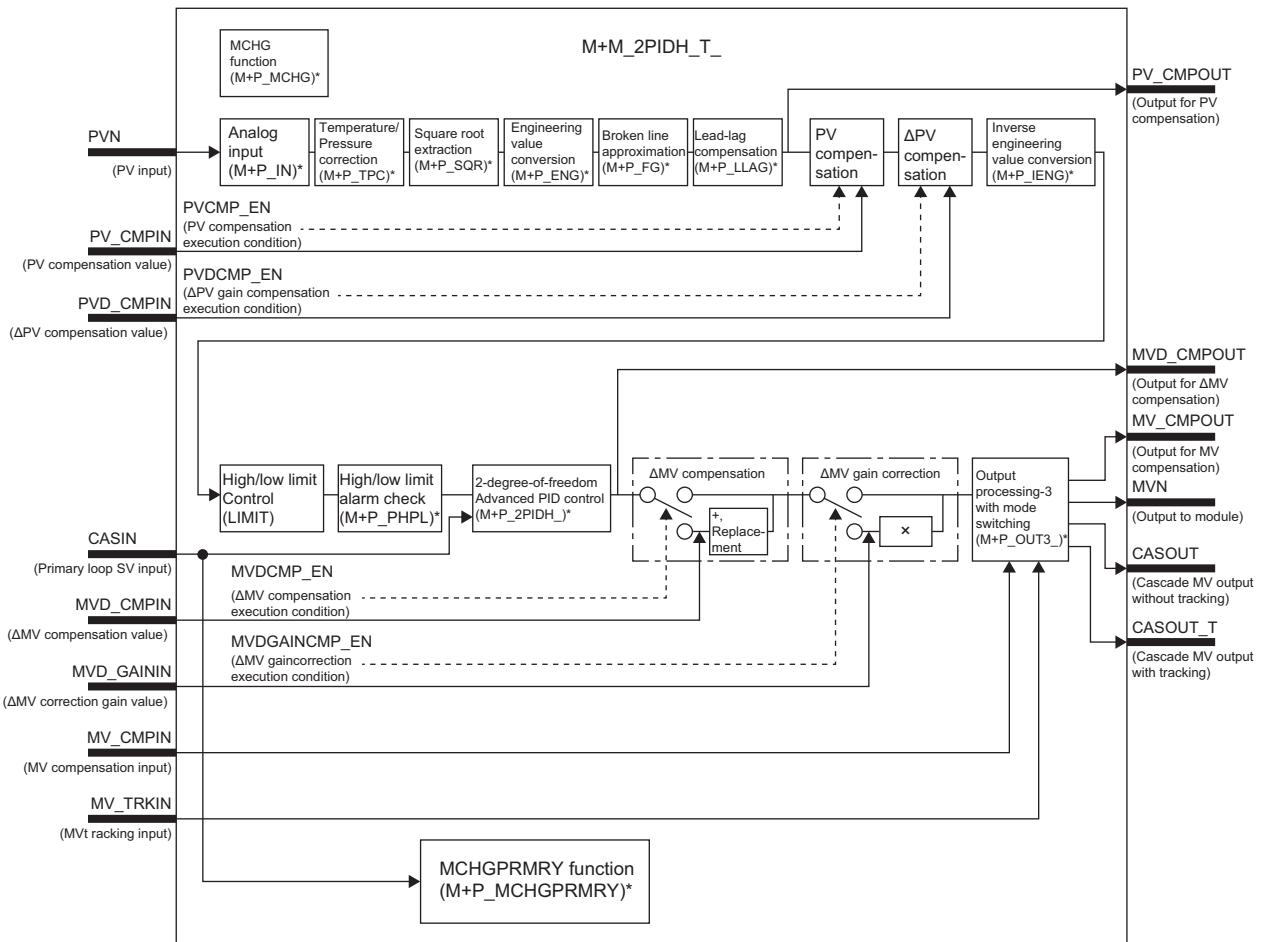
2PIDH

### Control mode

MAN	AUT	CAS*1	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

\*1 Transition to CASDR is possible.

## Block diagram



\*(M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

## Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN_T	Primary loop SV input (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
PV_CMPIN	PV compensation value	-999999 to 999999	Input variable	REAL
PVD_CMPIN	ΔPV compensation value	-999999 to 999999	Input variable	REAL
MVD_CMPIN	ΔMV compensation value (Unit: %)	-100 to 100[%]	Input variable	REAL
MVD_GAININ	ΔMV correction gain value	-999999 to 999999	Input variable	REAL
MV_CMPIN	MV compensation value (Unit: %)	-999999 to 999999[%]	Input variable	REAL
MV_TRKIN	MV tracking input (Unit: %)	0 to 100[%]	Input variable	REAL
MVN	Output to a module	OUT3_NMIN to OUT3_NMAX	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD
PV_CMPOUT	Output for PV compensation	-999999 to 999999	Output variable	REAL
MVD_CMPOUT	Output for ΔMV compensation (Unit: %)	-100 to 100[%]	Output variable	REAL
MV_CMPOUT	Output for MV compensation (Unit: %)	0 to 100[%]	Output variable	REAL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>15*6</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>2*3</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
TPC_SQR	Temperature Pressure Correction pattern	0: None 1: Square root extraction 2: Temperature correction + Square root extraction 3: Pressure correction + Square root extraction 4: Temperature/pressure correction + Square root extraction	0	User	INT
TPC_PVTEMP	Temperature Pressure Correction: Measured Temperature (Engineering Value)	-999999 to 999999	0.0	User	REAL
TPC_PVPRES	Temperature Pressure Correction: Measured Pressure (Engineering Value)	-999999 to 999999	0.0	User	REAL
TPC_TEMP	Temperature Pressure Correction: Design Temperature	-999999 to 999999	0.0	User	REAL
TPC_B1	Temperature Pressure Correction: Bias Temperature	-999999 to 999999	273.15	User	REAL
TPC_PRES	Temperature Pressure Correction: Design Pressure	-999999 to 999999	0.0	User	REAL
TPC_B2	Temperature Pressure Correction: Bias Pressure	-999999 to 999999	10332.0	User	REAL
SQR_OLC	Square Root Extraction: Output Low Cut-off Value	0 to 999999	0.0	User	REAL
SQR_K	Square Root Extraction: Coefficient	0 to 999999	10.0	User	REAL
SQR_DENSITY	Square Root Extraction: Density Correction Value	0 to 999999	1.0	User	REAL
FG_SN	Function Generator: Number of Points	0 to 48	0	User	INT
FG_X1 to FG_X48	Function Generator: Input coordinate (x-coordinate)	-999999 to 999999	0.0	User	REAL
FG_Y1 to FG_Y48	Function Generator: Output coordinate (y-coordinate)	-999999 to 999999	0.0	User	REAL
LLAG_EN	First Order Lag: Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
LLAG_T1	First Order Lag: Delay Time (Sec)	0 to 999999[s]	1.0	User	REAL
PVCOMP_EN	PV Compensation Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
PVCOMP_MODE	PV Compensation Mode	0: Addition 1: Replacement	0	User	INT
PVDCMP_EN	ΔPV Compensation Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
PID2H_MTD	Derivative Gain	0 to 9999	8.0	User	REAL
PID2H_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PID2H_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
PID2H_TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT

Variable name	Description	Recommended range	Initial value	Set by	Data type
PID2H_SVPTN_B0	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
PID2H_SVPTN_B1	Setting Value (SV) Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
PID2H_PVTRK_EN	PV Tracking Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
PID2H_ISTP	Integration Stop Signal	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
PID2H_DSTP	Derivation Stop Signal	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
PID2H_LMT_ISTP	Stop Integration, when MV variation rate limiter alarm occurred	TRUE: Stop FALSE: Not stop	FALSE	User	BOOL
PID2H_SVLMT_EN	SV High/Low Limiter	TRUE: Execute FALSE: Not execute	FALSE	User	BOOL
MVDCMP_EN	ΔMV Compensation Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
MVDCMP_MODE	ΔMV Compensation Mode	0: Addition 1: Replacement	0	User	INT
MVDGAINCMP_EN	ΔMV Gain Correction Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
OUT3_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
OUT3_MVCMP_EN	MV Compensation Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_MVCMP_MODE	MV Compensation Mode	0: Addition 1: Replacement	0	User	INT
OUT3_PREMV_EN	Preset MV Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_PREMV_V	Preset MV Value (Unit: %)	0 to 100[%]	0.0	User	REAL
OUT3_MVHLD_EN	MV Hold Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_MVTRK_EN	MV Tracking Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_STP_OTYPE	Output when loop or tag is stopped	0: Hold 1: Preset value	0	User	INT
OUT3_SEA_OTYPE	MV Output Selection when SEA is occurred	0: Hold 1: Preset MV output 2: Neither hold nor preset MV output is performed.	0	User	INT
OUT3_ARW_EX_EN	MV Value Instantaneous Pullback	TRUE: Execute FALSE: Not execute	FALSE	User	BOOL
OUT3_MVPH	MV Internal Operation High Limit Value (Unit: %)	MH to 999999[%]	100.0	User	REAL
OUT3_MVPL	MV Internal Operation Low Limit Value (Unit: %)	-999999 to ML[%]	0.0	User	REAL
OUT3_MVREV_EN	MV Reverse Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_FOTS_EN	Tight Shut/Full Open Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_MVFO	Output Value for Full Open (Unit: %)	100 to 125[%]	112.5	User	REAL
OUT3_MVTS	Output Value for Tight Shut (Unit: %)	-25 to 0[%]	-16.82	User	REAL
MCHG_BNAL_MODE <sup>2</sup> <sup>3</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT
MCHGPRMRY_SEA_EN <sup>2</sup>	When a sensor error occurs, switches the primary loop to MANUAL mode <sup>4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_OOA_EN <sup>2</sup>	When output open occurs, switches the primary loop to MANUAL mode <sup>4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_SPA_EN <sup>2</sup>	When a stop alarm occurs, switches the primary loop to MANUAL mode <sup>4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL



Variable name	Description	Recommended range	Initial value	Set by	Data type
MCHGPRMRY_TSTP_EN <sup>*2</sup>	When tag stop occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_CASCASDR_EN <sup>*2</sup>	When the control mode is changed (CAS/CASDR→AUT/MAN/CSV/CMV), switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*2 This public variable was added in version 1.070Y of the function block.

\*3 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*4 When TRUE (switch) is set, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*5 This public variable was added in version 1.085P of the function block.

\*6 If the version of the function block is lower than 1.085P, complies with the setting common to all tags.

## ■Public variable (others) <sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 6 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV 6: CASDR	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

📖 Page 823 2PIDH

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
TPC function	M+P_TPC	Page 94 M+P_TPC
SQR function	M+P_SQR	Page 111 M+P_SQR
ENG function	M+P_ENG	Page 90 M+P_ENG
FG function	M+P_FG	Page 82 M+P_FG
LLAG function	M+P_LLAG	Page 123 M+P_LLAG
IENG function	M+P_IENG	Page 92 M+P_IENG
LIMIT function	LIMIT	📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
2PIDH function	M+P_2PIDH_T_	Page 242 M+P_2PIDH_T_
OUT3 function	M+P_OUT3_	Page 198 M+P_OUT3_
MCHG function	M+P_MCHG	Page 378 M+P_MCHG
MCHGPRMRY function <sup>*1</sup>	M+P_MCHGPRMRY	Page 381 M+P_MCHGPRMRY

\*1 This function was added in version 1.070Y of the function block.

## ■PV compensation

This function block adds an externally-input compensation value to a process variable or replaces a process variable with an externally-input compensation value.

Condition		Processing result
PVCMP_EN = TRUE	PVCMP_MODE = 0 (addition)	IN + PV_CMPIN
	PVCMP_MODE = 1 (replacement)	PV_CMPIN
PVCMP_EN = FALSE		IN

IN: Input value (Process variable), PV\_CMPIN: Compensation value, PVCMP\_MODE: Compensation mode

## ■ $\Delta$ PV compensation

This function block adds a  $\Delta$ PV compensation value (PVD\_CMPIN) to an internal addition value ( $\Sigma$ PVD\_CMPIN) when PVDCMP\_EN is enabled.

$\Sigma$ PVD\_CMPIN is added to a process variable.

## ■ $\Delta$ MV compensation

This function block adds an externally-input compensation value to  $\Delta$ MV or replaces  $\Delta$ MV with an externally-input compensation value.

Condition		Processing result
MVDCMP_EN = TRUE	MVDCMP_MODE = 0 (addition)	IN + MVD_CMPIN
	MVDCMP_MODE = 1 (replacement)	MVD_CMPIN
MVDCMP_EN = FALSE		IN

IN: Input value ( $\Delta$ Manipulated value), MVD\_CMPIN: Compensation value, MVDCMP\_MODE: Compensation mode

## ■ $\Delta$ MV gain correction

This function block multiplies  $\Delta$ MV by a gain correction value.

Condition		Processing result
MVDGAINCMP_EN = TRUE		IN $\times$ MVD_GAININ
MVDGAINCMP_EN = FALSE		IN

IN: Input value ( $\Delta$ Manipulated value), MVD\_GAININ: Gain correction value

## Operation error

Error code (SD0)	Description
3402H	Operation result of input data, operation constants, tag data, or inside of function blocks is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

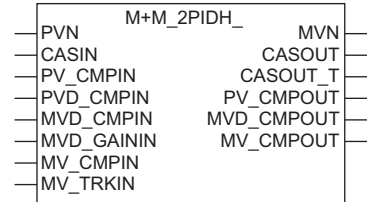
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.10 2-degree-of-freedom Advanced PID Control (Disable Tracking for primary loop) (M+M\_2PIDH\_)

## M+M\_2PIDH\_

This FB performs two-degree-of-freedom PID control combining the functions of M+P\_IN, M+P\_PHPL, M+P\_2PIDH\_, and M+P\_OUT3\_ into a single FB with PV/MV correction.

### FBD/LD



### Applicable tag type

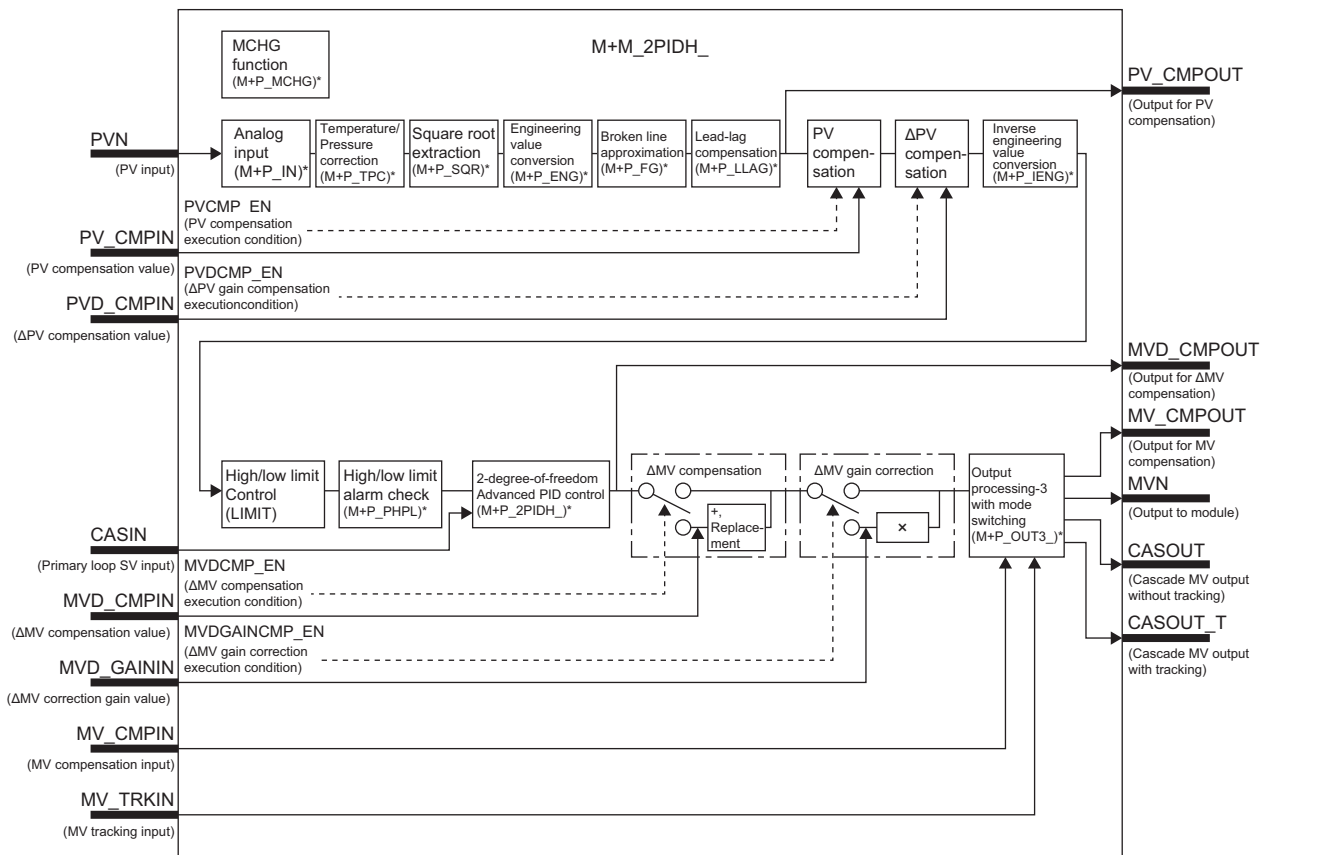
2PIDH

### Control mode

MAN	AUT	CAS*1	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

\*1 Transition to CASDR is possible.

### Block diagram



\*(M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

## Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN	Primary loop SV input (Unit: %)	0 to 100[%]	Input variable	REAL
PV_CMPIN	PV compensation value	-999999 to 999999	Input variable	REAL
PVD_CMPIN	$\Delta$ PV compensation value	-999999 to 999999	Input variable	REAL
MVD_CMPIN	$\Delta$ MV compensation value (Unit: %)	-100 to 100[%]	Input variable	REAL
MVD_GAININ	$\Delta$ MV correction gain value	-999999 to 999999	Input variable	REAL
MV_CMPIN	MV compensation value (Unit: %)	-999999 to 999999[%]	Input variable	REAL
MV_TRKIN	MV tracking input (Unit: %)	0 to 100[%]	Input variable	REAL
MVN	Output to a module	OUT3_NMIN to OUT3_NMAX	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD
PV_CMPOUT	Output for PV compensation	-999999 to 999999	Output variable	REAL
MVD_CMPOUT	Output for $\Delta$ MV compensation (Unit: %)	-100 to 100[%]	Output variable	REAL
MV_CMPOUT	Output for MV compensation (Unit: %)	0 to 100[%]	Output variable	REAL

### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*2</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*1</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
TPC_SQR	Temperature Pressure Correction pattern	0: None 1: Square root extraction 2: Temperature correction + Square root extraction 3: Pressure correction + Square root extraction 4: Temperature/pressure correction + Square root extraction	0	User	INT
TPC_PVTEMP	Temperature Pressure Correction: Measured Temperature (Engineering Value)	-999999 to 999999	0.0	User	REAL
TPC_PVPRES	Temperature Pressure Correction: Measured Pressure (Engineering Value)	-999999 to 999999	0.0	User	REAL
TPC_TEMP	Temperature Pressure Correction: Design Temperature	-999999 to 999999	0.0	User	REAL
TPC_B1	Temperature Pressure Correction: Bias Temperature	-999999 to 999999	273.15	User	REAL
TPC_PRES	Temperature Pressure Correction: Design Pressure	-999999 to 999999	0.0	User	REAL
TPC_B2	Temperature Pressure Correction: Bias Pressure	-999999 to 999999	10332.0	User	REAL
SQR_OLC	Square Root Extraction: Output Low Cut-off Value	0 to 999999	0.0	User	REAL
SQR_K	Square Root Extraction: Coefficient	0 to 999999	10.0	User	REAL

Variable name	Description	Recommended range	Initial value	Set by	Data type
SQR_DENSITY	Square Root Extraction: Density Correction Value	0 to 999999	1.0	User	REAL
FG_SN	Function Generator: Number of Points	0 to 48	0	User	INT
FG_X1 to FG_X48	Function Generator: Input coordinate (x-coordinate)	-999999 to 999999	0.0	User	REAL
FG_Y1 to FG_Y48	Function Generator: Output coordinate (y-coordinate)	-999999 to 999999	0.0	User	REAL
LLAG_EN	First Order Lag: Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
LLAG_T1	First Order Lag: Delay Time (Sec)	0 to 999999[s]	1.0	User	REAL
PVCOMP_EN	PV Compensation Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
PVCOMP_MODE	PV Compensation Mode	0: Addition 1: Replacement	0	User	INT
PVDCMP_EN	$\Delta$ PV Compensation Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
PID2H_MTD	Derivative Gain	0 to 9999	8.0	User	REAL
PID2H_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PID2H_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
PID2H_SVPTN_B0	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
PID2H_PVTRK_EN	PV Tracking Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
PID2H_ISTP	Integration Stop Signal	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
PID2H_DSTP	Derivation Stop Signal	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
PID2H_LMT_ISTP	Stop Integration, when MV variation rate limiter alarm occurred	TRUE: Stop FALSE: Not stop	FALSE	User	BOOL
PID2H_SVLMT_EN	SV High/Low Limiter	TRUE: Execute FALSE: Not execute	FALSE	User	BOOL
MVDCMP_EN	$\Delta$ MV Compensation Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
MVDCMP_MODE	$\Delta$ MV Compensation Mode	0: Addition 1: Replacement	0	User	INT
MVDGAINCMP_EN	$\Delta$ MV Gain Correction Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
OUT3_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
OUT3_MVCOMP_EN	MV Compensation Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_MVCOMP_MODE	MV Compensation Mode	0: Addition 1: Replacement	0	User	INT
OUT3_PREMV_EN	Preset MV Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_PREMV_V	Preset MV Value (Unit: %)	0 to 100[%]	0.0	User	REAL
OUT3_MVHLD_EN	MV Hold Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_MVTRK_EN	MV Tracking Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_STP_OTYPE	Output when loop or tag is stopped	0: Hold 1: Preset value	0	User	INT
OUT3_SEA_OTYPE	MV Output Selection when SEA is occurred	0: Hold 1: Preset MV output 2: Neither hold nor preset MV output is performed.	0	User	INT
OUT3_ARW_EX_EN	MV Value Instantaneous Pullback	TRUE: Execute FALSE: Not execute	FALSE	User	BOOL
OUT3_MVPH	MV Internal Operation High Limit Value (Unit: %)	MH to 999999[%]	100.0	User	REAL

Variable name	Description	Recommended range	Initial value	Set by	Data type
OUT3_MVPL	MV Internal Operation Low Limit Value (Unit: %)	-999999 to ML[%]	0.0	User	REAL
OUT3_MVREV_EN	MV Reverse Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_FOTS_EN	Tight Shut/Full Open Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
OUT3_MVFO	Output Value for Full Open (Unit: %)	100 to 125[%]	112.5	User	REAL
OUT3_MVTS	Output Value for Tight Shut (Unit: %)	-25 to 0[%]	-16.82	User	REAL
MCHG_BNAL_MODE <sup>*1</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT

\*1 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

\*2 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

### ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

☞ Page 823 2PIDH

### ■Public variable (others) <sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL


- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 6 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV 6: CASDR	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
TPC function	M+P_TPC	Page 94 M+P_TPC
SQR function	M+P_SQR	Page 111 M+P_SQR
ENG function	M+P_ENG	Page 90 M+P_ENG
FG function	M+P_FG	Page 82 M+P_FG
LLAG function	M+P_LLAG	Page 123 M+P_LLAG
IENG function	M+P_IENG	Page 92 M+P_IENG
LIMIT function	LIMIT	 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
2PIDH function	M+P_2PIDH_	Page 262 M+P_2PIDH_
OUT3 function	M+P_OUT3_	Page 198 M+P_OUT3_
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

### ■PV compensation

This function block adds an externally-input compensation value to a process variable or replaces a process variable with an externally-input compensation value.

Condition		Processing result
PVCMP_EN = TRUE	PVCMP_MODE = 0 (addition)	IN + PV_CMPIN
	PVCMP_MODE = 1 (replacement)	PV_CMPIN
PVCMP_EN = FALSE		IN

IN: Input value (Process variable), PV\_CMPIN: Compensation value, PVCMP\_MODE: Compensation mode

### ■ $\Delta$ PV compensation

This function block adds a  $\Delta$ PV compensation value (PVD\_CMPIN) to an internal addition value ( $\Sigma$ PVD\_CMPIN) when PVDCMP\_EN is enabled.

$\Sigma$ PVD\_CMPIN is added to a process variable.

### ■ $\Delta$ MV compensation

This function block adds an externally-input compensation value to  $\Delta$ MV or replaces  $\Delta$ MV with an externally-input compensation value.

Condition		Processing result
MVDCMP_EN = TRUE	MVDCMP_MODE = 0 (addition)	IN + MVD_CMPIN
	MVDCMP_MODE = 1 (replacement)	MVD_CMPIN
MVDCMP_EN = FALSE		IN

IN: Input value ( $\Delta$ Manipulated value), MVD\_CMPIN: Compensation value, MVDCMP\_MODE: Compensation mode

### ■ $\Delta$ MV gain correction

This function block multiplies  $\Delta$ MV by a gain correction value.

Condition	Processing result
MVDGAINCMP_EN = TRUE	IN $\times$ MVD_GAININ
MVDGAINCMP_EN = FALSE	IN

IN: Input value ( $\Delta$ Manipulated value), MVD\_GAININ: Gain correction value



## Operation error

Error code (SD0)	Description
3402H	Operation result of input data, operation constants, tag data, or inside of function blocks is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

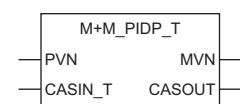
# 17.11 Position Type PID Control (Enable Tracking for primary loop/Disable Tracking from secondary loop) (M+M\_PIDP\_T)

## M+M\_PIDP\_T

This FB performs position type PID control combining the functions of M+P\_IN, M+P\_PHPL, and M+P\_PIDP\_T into a single FB.

The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.

### FBD/LD



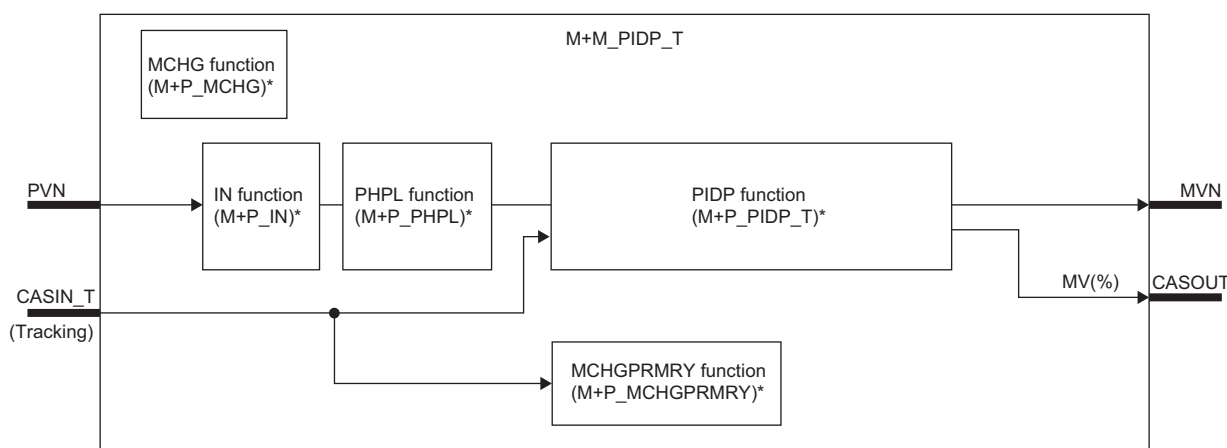
### Applicable tag type

PIDP

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN_T	Primary loop SV input (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVN	Output to a module	PIDP_NMIN to PIDP_NMAX	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*5*6</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*2*3</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
PIDP_MTD	Derivative Gain	0 to 9999	8.0	User	REAL
PIDP_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PIDP_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
PIDP_TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
PIDP_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
PIDP_SVPTN_B1	Setting Value (SV) Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
PIDP_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
PIDP_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
MCHG_BNAL_MODE <sup>*2*3</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT
MCHGPRMRY_SEA_EN <sup>*2</sup>	When a sensor error occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_OOA_EN <sup>*2</sup>	When output open occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_SPA_EN <sup>*2</sup>	When a stop alarm occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_CASCASDR_EN <sup>*2</sup>	When the control mode is changed (CAS→AUT/MAN/CSV/CMV), switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*2 This public variable was added in version 1.070Y of the function block.

\*3 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*4 When TRUE (switch) is set, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*5 This public variable was added in version 1.085P of the function block.

\*6 If the version of the function block is lower than 1.085P, complies with the setting common to all tags.

## ■ Public variable (others)<sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■ Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

📖 Page 829 PIDP

### Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
PIDP function	M+P_PIDP_T	Page 269 M+P_PIDP_T
MCHG function	M+P_MCHG	Page 378 M+P_MCHG
MCHGPRMRY function <sup>*1</sup>	M+P_MCHGPRMRY	Page 381 M+P_MCHGPRMRY

\*1 This function was added in version 1.070Y of the function block.

### Operation error

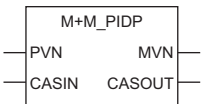
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.12 Position Type PID Control (Disable Tracking for primary loop/Disable Tracking from secondary loop) (M+M\_PIDP)

## M+M\_PIDP

This FB performs position type PID control combining the functions of M+P\_IN, M+P\_PHPL, and M+P\_PIDP into a single FB.

### FBD/LD



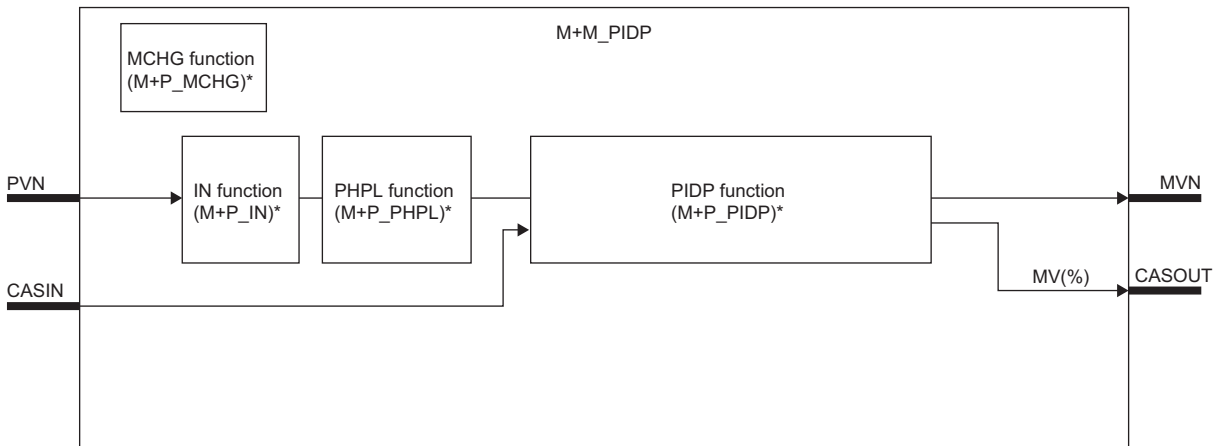
### Applicable tag type

PIDP

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN	Primary loop SV input (Unit: %)	0 to 100[%]	Input variable	REAL
MVN	Output to a module	PIDP_NMIN to PIDP_NMAX	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL

## ■Public variable (operation constant)

- Operation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*2</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*1</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
PIDP_MTD	Derivative Gain	0 to 9999	8.0	User	REAL
PIDP_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PIDP_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
PIDP_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
PIDP_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
PIDP_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
MCHG_BNAL_MODE <sup>*1</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT

\*1 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

\*2 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

## ■Public variable (others) <sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

📄 Page 829 PIDP

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
PIDP function	M+P_PIDP	Page 276 M+P_PIDP
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

## Operation error

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

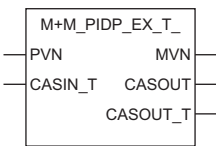
# 17.13 Position Type PID Control (Enable Tracking for primary loop/Enable Tracking from secondary loop) (M+M\_PIDP\_EX\_T\_)

## M+M\_PIDP\_EX\_T\_

This FB performs position type PID control combining the functions of M+P\_IN, M+P\_PHPL, and M+P\_PIDP\_EX\_T\_ into a single FB. It also allows manipulated value bumpless switching at a change of control mode.

In the cascade connection, it allows tracking of the primary loop or tracking from the secondary loop. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.

### FBD/LD



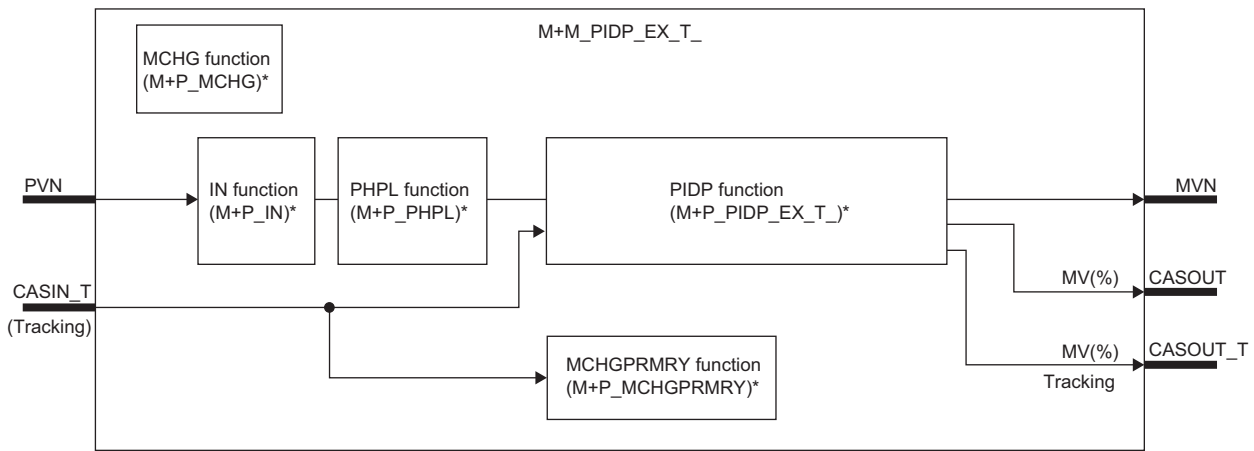
### Applicable tag type

PIDP

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN_T	Primary loop SV input (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVN	Output to a module	PIDP_NMIN to PIDP_NMAX	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD



## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*5*6</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*2*3</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
PIDP_MTD	Derivative Gain	0 to 9999	8.0	User	REAL
PIDP_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PIDP_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
PIDP_TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
PIDP_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
PIDP_SVPTN_B1	Setting Value (SV) Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
PIDP_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
PIDP_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
MCHG_BNAL_MODE <sup>*2*3</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT
MCHGPRMRY_SEA_EN <sup>*2</sup>	When a sensor error occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_OOA_EN <sup>*2</sup>	When output open occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_SPA_EN <sup>*2</sup>	When a stop alarm occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_CASCASDR_EN <sup>*2</sup>	When the control mode is changed (CAS→AUT/MAN/CSV/CMV), switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*2 This public variable was added in version 1.070Y of the function block.

\*3 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*4 When TRUE (switch) is set, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*5 This public variable was added in version 1.085P of the function block.

\*6 If the version of the function block is lower than 1.085P, complies with the setting common to all tags.

## ■ Public variable (others)<sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■ Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

📖 Page 829 PIDP

### Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
PIDP function	M+P_PIDP_EX_T_	Page 283 M+P_PIDP_EX_T_
MCHG function	M+P_MCHG	Page 378 M+P_MCHG
MCHGPRMRY function <sup>*1</sup>	M+P_MCHGPRMRY	Page 381 M+P_MCHGPRMRY

\*1 This function was added in version 1.070Y of the function block.

### Operation error

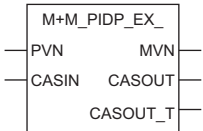
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.14 Position Type PID Control (Disable Tracking for primary loop/Enable Tracking from secondary loop) (M+M\_PIDP\_EX\_)

## M+M\_PIDP\_EX\_

This FB performs position type PID control combining the functions of M+P\_IN, M+P\_PHPL, and M+P\_PIDP\_EX\_ into a single FB. It also allows manipulated value bumpless switching at a change of control mode. In the cascade connection, it allows tracking from the secondary loop.

### FBD/LD



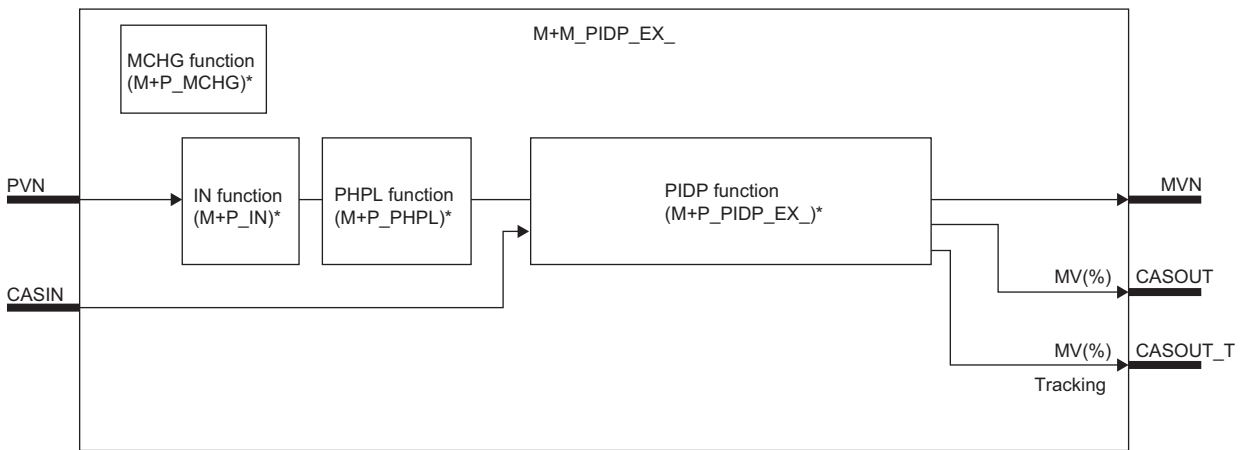
### Applicable tag type

PIDP

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN	Primary loop SV input (Unit: %)	0 to 100[%]	Input variable	REAL
MVN	Output to a module	PIDP_NMIN to PIDP_NMAX	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

## ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*2</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*1</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
PIDP_MTD	Derivative Gain	0 to 9999	8.0	User	REAL
PIDP_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PIDP_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
PIDP_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
PIDP_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
PIDP_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
MCHG_BNAL_MODE <sup>*1</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT

\*1 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

\*2 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

## ■Public variable (others) <sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

📄 Page 829 PIDP

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
PIDP function	M+P_PIDP_EX_	Page 291 M+P_PIDP_EX_
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

## Operation error

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

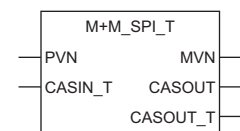
# 17.15 Sample PI Control (Enable Tracking for primary loop) (M+M\_SPI\_T)

## M+M\_SPI\_T

This FB performs sample PI control combining the functions of M+P\_IN, M+P\_PHPL, M+P\_SPI\_T, and M+P\_OUT1 into a single FB.

The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.

### FBD/LD



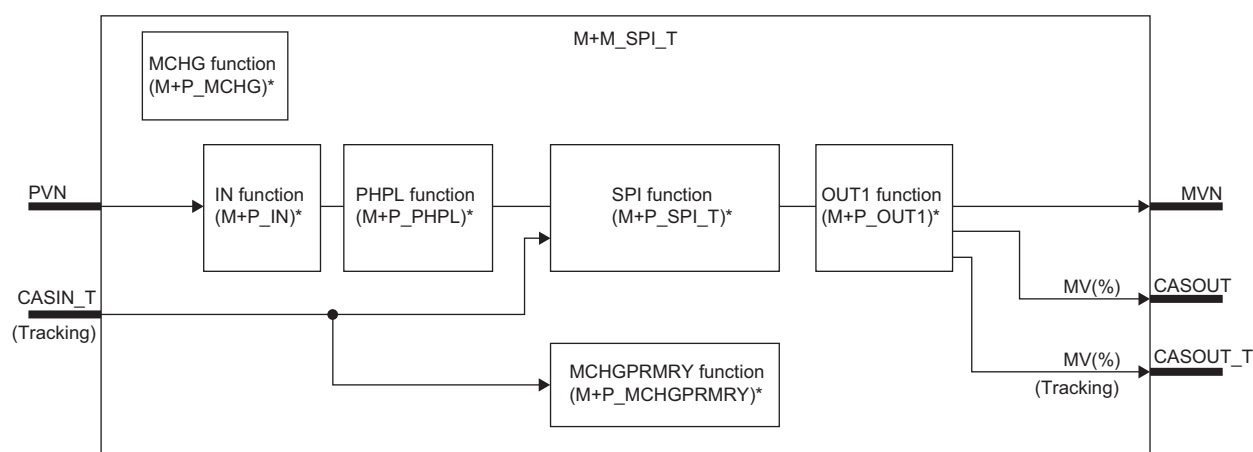
### Applicable tag type

SPI

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN_T	Primary loop SV input (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVN	Output to a module	OUT1_NMIN to OUT1_NMAX	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*5*6</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*2*3</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
SPI_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
SPI_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
SPI_TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
SPI_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SPI_SVPTN_B1	Setting Value (SV) Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
OUT1_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
OUT1_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
OUT1_SEA_OTYPE <sup>*5*6</sup>	Hold processing selection for output function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT
MCHG_BNAL_MODE <sup>*2*3</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT
MCHGPRMRY_SEA_EN <sup>*2</sup>	When a sensor error occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_OOA_EN <sup>*2</sup>	When output open occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_SPA_EN <sup>*2</sup>	When a stop alarm occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_CASCASDR_EN <sup>*2</sup>	When the control mode is changed (CAS→AUT/MAN/CSV/CMV), switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*2 This public variable was added in version 1.070Y of the function block.

\*3 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*4 When TRUE (switch) is set, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*5 This public variable was added in version 1.085P of the function block.

\*6 If the version of the function block is lower than 1.085P, complies with the setting common to all tags.

## ■Public variable (others) \*1

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

☞ Page 833 SPI

### Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
SPI function	M+P_SPI_T	Page 299 M+P_SPI_T
OUT1 function	M+P_OUT1	Page 189 M+P_OUT1
MCHG function	M+P_MCHG	Page 378 M+P_MCHG
MCHGPRMRY function*1	M+P_MCHGPRMRY	Page 381 M+P_MCHGPRMRY

\*1 This function was added in version 1.070Y of the function block.

### Operation error

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

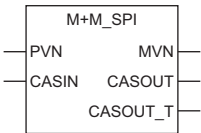


# 17.16 Sample PI Control (Disable Tracking for primary loop) (M+M\_SPI)

## M+M\_SPI

This FB performs sample PI control combining the functions of M+P\_IN, M+P\_PHPL, M+P\_SPI, and M+P\_OUT1 into a single FB.

### FBD/LD



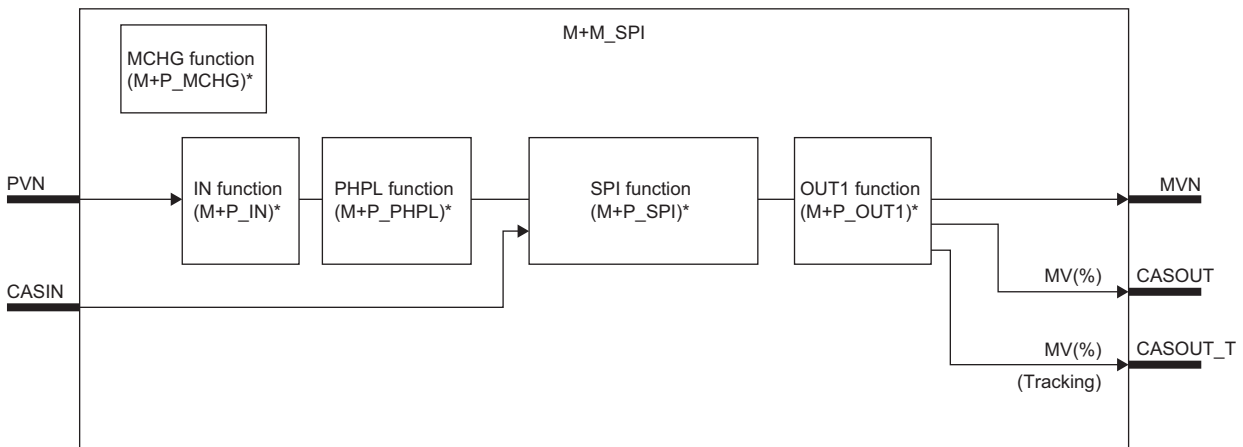
### Applicable tag type

SPI

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN	Primary loop SV input (Unit: %)	0 to 100[%]	Input variable	REAL
MVN	Output to a module	OUT1_NMIN to OUT1_NMAX	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

## ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*2</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*1</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
SPI_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
SPI_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
SPI_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
OUT1_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
OUT1_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
OUT1_SEA_OTYPE <sup>*2</sup>	Hold processing selection for output function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT
MCHG_BNAL_MODE <sup>*1</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT

\*1 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

\*2 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

## ■Public variable (others) <sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

☞ Page 833 SPI

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
SPI function	M+P_SPI	Page 305 M+P_SPI
OUT1 function	M+P_OUT1	Page 189 M+P_OUT1
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

## Operation error

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

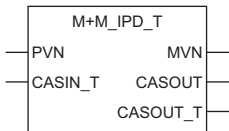
# 17.17 I-PD Control (Enable Tracking for primary loop) (M+M\_IPD\_T)

## M+M\_IPD\_T

This FB performs I-PD control combining the functions of M+P\_IN, M+P\_PHPL, M+P\_IPD\_T, and M+P\_OUT1 into a single FB.

The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.

### FBD/LD



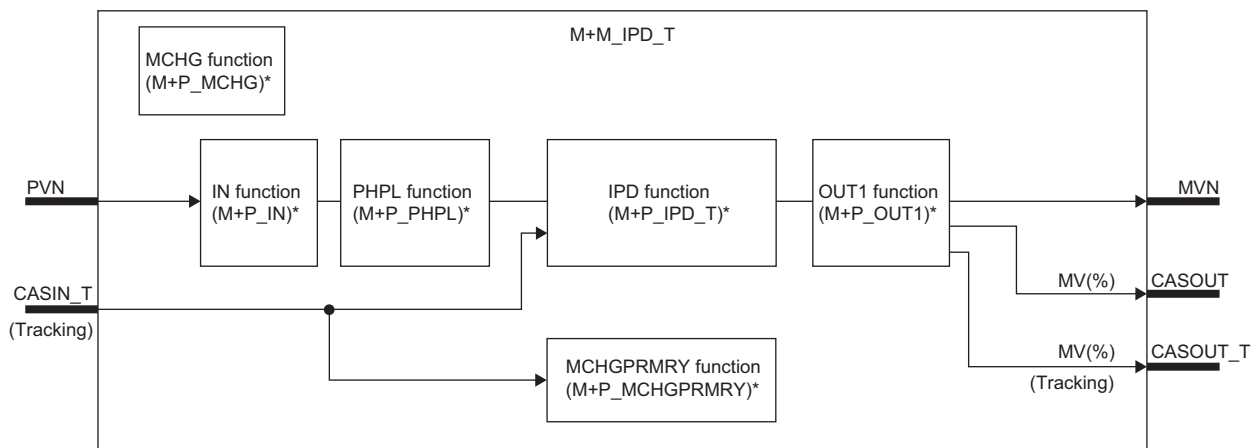
### Applicable tag type

IPD

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN_T	Primary loop SV input (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVN	Output to a module	OUT1_NMIN to OUT1_NMAX	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*5*6</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*2*3</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
IPD_MTD	Derivative Gain	0 to 9999	8.0	User	REAL
IPD_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
IPD_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
IPD_TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
IPD_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
IPD_SVPTN_B1	Setting Value (SV) Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
OUT1_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
OUT1_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
OUT1_SEA_OTYPE <sup>*5*6</sup>	Hold processing selection for output function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT
MCHG_BNAL_MODE <sup>*2*3</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT
MCHGPRMRY_SEA_EN <sup>*2</sup>	When a sensor error occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_OOA_EN <sup>*2</sup>	When output open occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_SPA_EN <sup>*2</sup>	When a stop alarm occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_CASCASDR_EN <sup>*2</sup>	When the control mode is changed (CAS→AUT/MAN/CSV/CMV), switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*2 This public variable was added in version 1.070Y of the function block.

\*3 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*4 When TRUE (switch) is set, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*5 This public variable was added in version 1.085P of the function block.

\*6 If the version of the function block is lower than 1.085P, complies with the setting common to all tags.

## ■ Public variable (others)<sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■ Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

☞ Page 837 IPD

### Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
IPD function	M+P_IPD_T	Page 310 M+P_IPD_T
OUT1 function	M+P_OUT1	Page 189 M+P_OUT1
MCHG function	M+P_MCHG	Page 378 M+P_MCHG
MCHGPRMRY function <sup>*1</sup>	M+P_MCHGPRMRY	Page 381 M+P_MCHGPRMRY

\*1 This function was added in version 1.070Y of the function block.

### Operation error

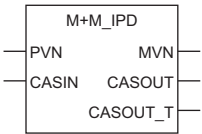
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.18 I-PD Control (Disable Tracking for primary loop) (M+M\_IPD)

## M+M\_IPD

This FB performs I-PD control combining the functions of M+P\_IN, M+P\_PHPL, M+P\_IPD, and M+P\_OUT1 into a single FB.

### FBD/LD



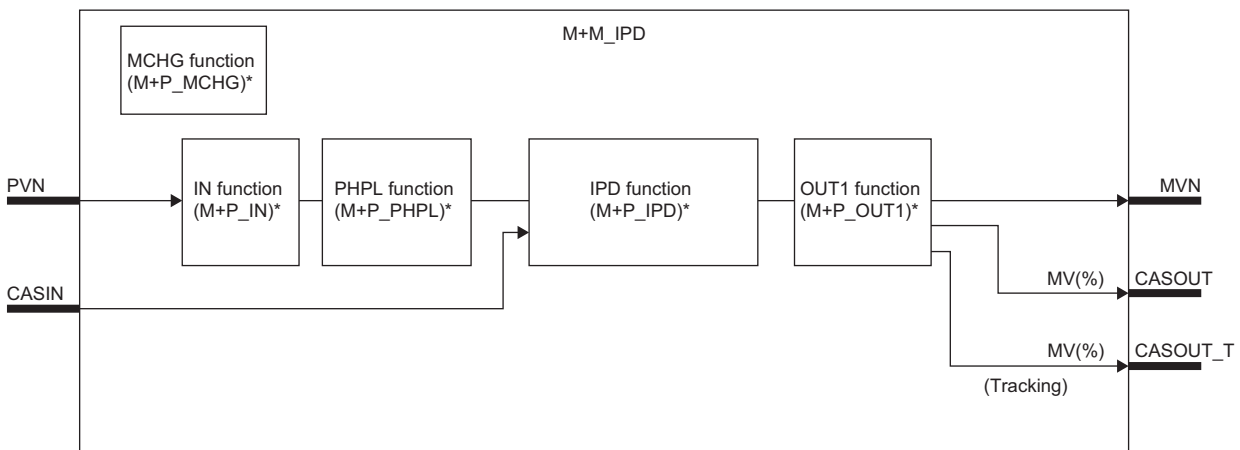
### Applicable tag type

IPD

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN	Primary loop SV input (Unit: %)	0 to 100[%]	Input variable	REAL
MVN	Output to a module	OUT1_NMIN to OUT1_NMAX	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

## ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*2</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*1</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
IPD_MTD	Derivative Gain	0 to 9999	8.0	User	REAL
IPD_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
IPD_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
IPD_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
OUT1_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
OUT1_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
OUT1_SEA_OTYPE <sup>*2</sup>	Hold processing selection for output function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT
MCHG_BNAL_MODE <sup>*1</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT

\*1 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

\*2 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

## ■Public variable (others) <sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.



## ■ Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

☞ Page 837 IPD

### Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
IPD function	M+P_IPD	Page 316 M+P_IPD
OUT1 function	M+P_OUT1	Page 189 M+P_OUT1
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

### Operation error

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

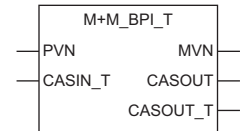
# 17.19 Blend PI Control (Enable Tracking for primary loop) (M+M\_BPI\_T)

## M+M\_BPI\_T

This FB performs blend PI control combining the functions of M+P\_IN, M+P\_PHPL, M+P\_BPI\_T, and M+P\_OUT1 into a single FB.

The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.

### FBD/LD



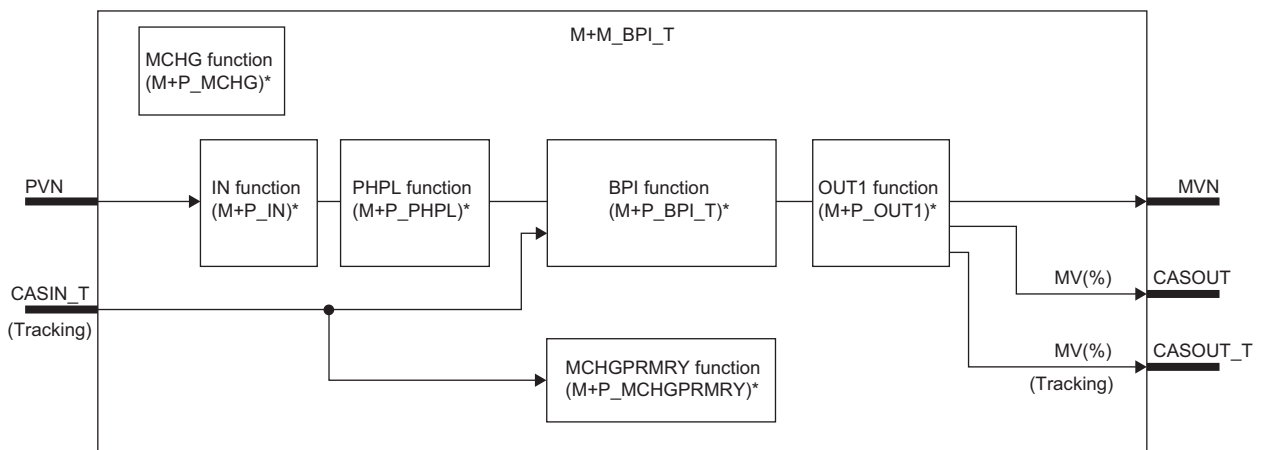
### Applicable tag type

BPI

### Control mode

MAN	AUT	CAS	CMV	CSV
○	○	○	○	○

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN_T	Primary loop SV input (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVN	Output to a module	OUT1_NMIN to OUT1_NMAX	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*5*6</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*2*3</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
BPI_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
BPI_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
BPI_TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
BPI_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
BPI_SVPTN_B1	Setting Value (SV) Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
BPI_RST_SDV_ON_CHGMODE	Reset DV Cumulative Value in Control Mode Change	TRUE: DV cumulative value (SDV) reset at control mode change (from MAN/CMV to AUT/CAS/CSV) FALSE: DV cumulative value (SDV) not reset	FALSE	User	BOOL
OUT1_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
OUT1_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
OUT1_SEA_OTYPE <sup>*5*6</sup>	Hold processing selection for output function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT
MCHG_BNAL_MODE <sup>*2*3</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT
MCHGPRMRY_SEA_EN <sup>*2</sup>	When a sensor error occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_OOA_EN <sup>*2</sup>	When output open occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_SPA_EN <sup>*2</sup>	When a stop alarm occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_CASCASDR_EN <sup>*2</sup>	When the control mode is changed (CAS→AUT/MAN/CSV/CMV), switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*2 This public variable was added in version 1.070Y of the function block.

\*3 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*4 When TRUE (switch) is set, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*5 This public variable was added in version 1.085P of the function block.

\*6 If the version of the function block is lower than 1.085P, complies with the setting common to all tags.

## ■Public variable (others) \*1

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

- Operation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
BPI_RST_SDV	Reset DV Cumulative Value	TRUE, FALSE FALSE → TRUE: DV cumulative value (SDV) reset	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

☞ Page 841 BPI

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
BPI function	M+P_BPI_T	Page 321 M+P_BPI_T
OUT1 function	M+P_OUT1	Page 189 M+P_OUT1
MCHG function	M+P_MCHG	Page 378 M+P_MCHG
MCHGPRMRY function*1	M+P_MCHGPRMRY	Page 381 M+P_MCHGPRMRY

\*1 This function was added in version 1.070Y of the function block.

## Operation error

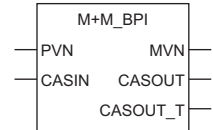
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.20 Blend PI Control (Disable Tracking for primary loop) (M+M\_BPI)

## M+M\_BPI

This FB performs blend PI control combining the functions of M+P\_IN, M+P\_PHPL, M+P\_BPI, and M+P\_OUT1 into a single FB.

### FBD/LD



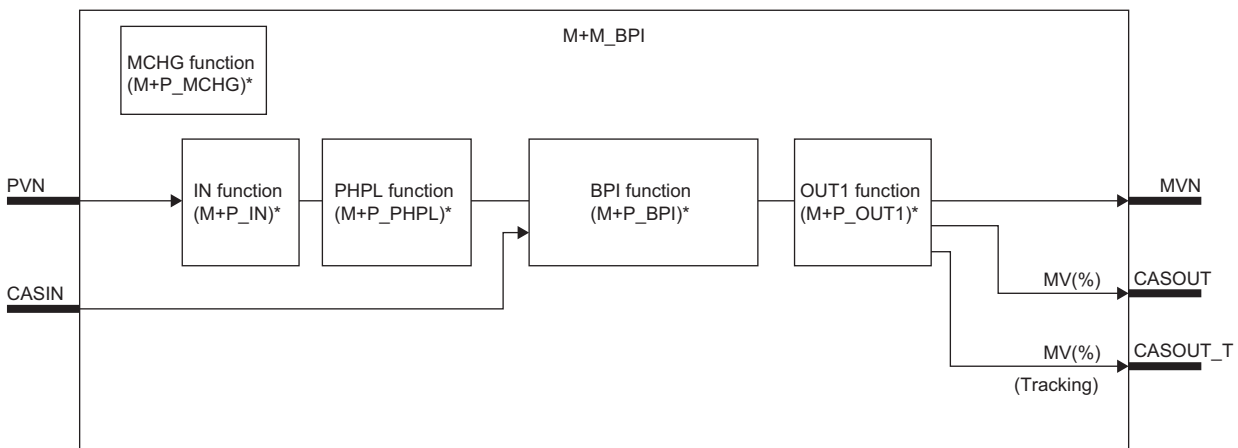
### Applicable tag type

BPI

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN	Primary loop SV input (Unit: %)	0 to 100[%]	Input variable	REAL
MVN	Output to a module	OUT1_NMIN to OUT1_NMAX	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade MV output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

## ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*2</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*1</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
BPI_DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
BPI_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
BPI_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
BPI_RST_SDV_ON_CHGMODE	Reset DV Cumulative Value in Control Mode Change	TRUE: DV cumulative value (SDV) reset at control mode change (from MAN/CMV to AUT/CAS/CSV) FALSE: DV cumulative value (SDV) not reset	FALSE	User	BOOL
OUT1_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
OUT1_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
OUT1_SEA_OTYPE <sup>*2</sup>	Hold processing selection for output function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT
MCHG_BNAL_MODE <sup>*1</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT

\*1 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

\*2 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

■ **Public variable (others) \*1**

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: GSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

- Operation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
BPI_RST_SDV	Reset DV Cumulative Value	TRUE, FALSE FALSE → TRUE: DV cumulative value (SDV) reset	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

■ **Tag data**

For details on tag data that is read or written with this tag FB, refer to the following.

☞ Page 841 BPI

**Processing details**

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
BPI function	M+P_BPI	Page 326 M+P_BPI
OUT1 function	M+P_OUT1	Page 189 M+P_OUT1
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

**Operation error**

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.21 Ratio Control (Enable Tracking for primary loop) (M+M\_R\_T)

## M+M\_R\_T

This FB performs ratio control combining the functions of M+P\_IN, M+P\_PHPL, M+P\_R\_T, and M+P\_OUT2 into a single FB. The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.

### FBD/LD



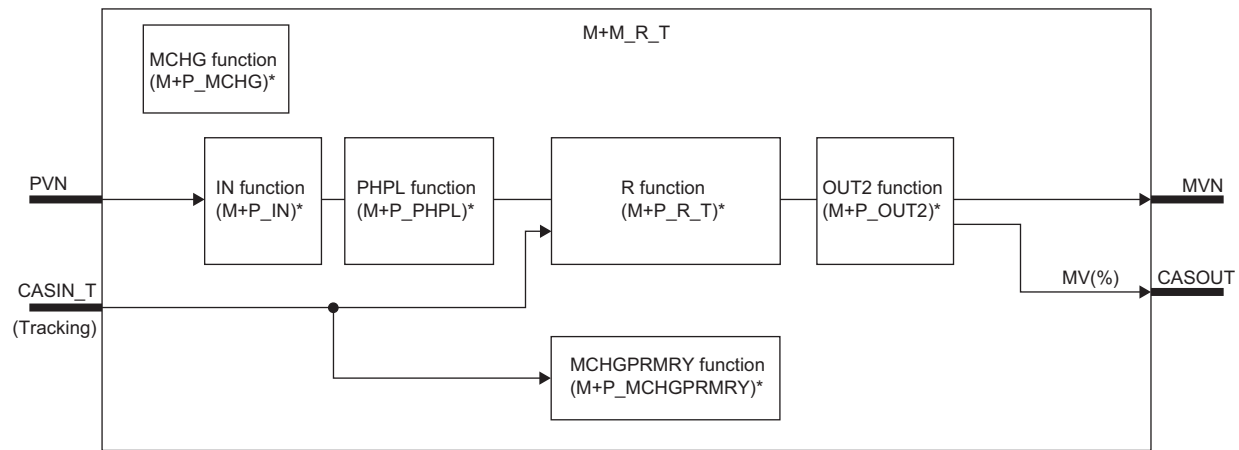
### Applicable tag type

R

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN_T	Primary loop SV input (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVN	Output to a module	OUT2_NMIN to OUT2_NAMX	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL



■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*5*6</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*2*3</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
R_TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
R_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
R_SVPTN_B1	Setting Value (SV) Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
OUT2_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
OUT2_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
OUT2_SEA_OTYPE <sup>*5*6</sup>	Hold processing selection for output function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT
MCHG_BNAL_MODE <sup>*2*3</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT
MCHGPRMRY_SEA_EN <sup>*2</sup>	When a sensor error occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_OOA_EN <sup>*2</sup>	When output open occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_SPA_EN <sup>*2</sup>	When a stop alarm occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_CASCASDR_EN <sup>*2</sup>	When the control mode is changed (CAS→AUT/MAN/CSV/CMV), switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*2 This public variable was added in version 1.070Y of the function block.

\*3 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*4 When TRUE (switch) is set, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*5 This public variable was added in version 1.085P of the function block.

\*6 If the version of the function block is lower than 1.085P, complies with the setting common to all tags.

## ■Public variable (others)<sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

☞ Page 845 R

### Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
R function	M+P_R_T	Page 223 M+P_R_T
OUT2 function	M+P_OUT2	Page 194 M+P_OUT2
MCHG function	M+P_MCHG	Page 378 M+P_MCHG
MCHGPRMRY function <sup>*1</sup>	M+P_MCHGPRMRY	Page 381 M+P_MCHGPRMRY

\*1 This function was added in version 1.070Y of the function block.

### Operation error

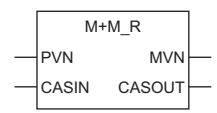
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.22 Ratio Control (Disable Tracking for primary loop) (M+M\_R)

## M+M\_R

This FB performs ratio control combining the functions of M+P\_IN, M+P\_PHPL, M+P\_R, and M+P\_OUT2 into a single FB.

### FBD/LD



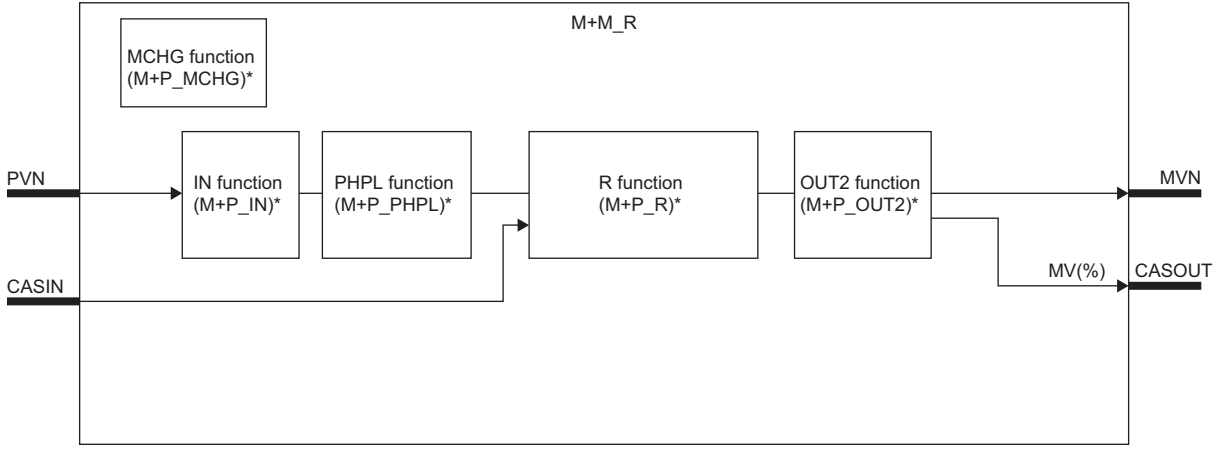
### Applicable tag type

R

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN	Primary loop SV input (Unit: %)	0 to 100[%]	Input variable	REAL
MVN	Output to a module	OUT2_NMIN to OUT2_NMAX	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL

## ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*2</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*1</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
R_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
OUT2_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
OUT2_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
OUT2_SEA_OTYPE <sup>*2</sup>	Hold processing selection for output function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold output 2: Continue operation	0	User	INT
MCHG_BNAL_MODE <sup>*1</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT

\*1 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

\*2 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

## ■Public variable (others) <sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL
SIMOUT	Simulation Output	NMIN to NMAX	0.0	System	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

📖 Page 845 R

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
R function	M+P_R	Page 227 M+P_R
OUT2 function	M+P_OUT2	Page 194 M+P_OUT2
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

## Operation error

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

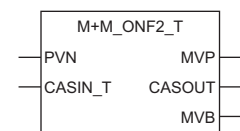
# 17.23 2 position ON/OFF Control (Enable Tracking for primary loop) (M+M\_ONF2\_T)

## M+M\_ONF2\_T

This FB performs two-position (on/off) control combining the functions of M+P\_IN, M+P\_PHPL, and M+P\_ONF2\_T into a single FB.

The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.

### FBD/LD



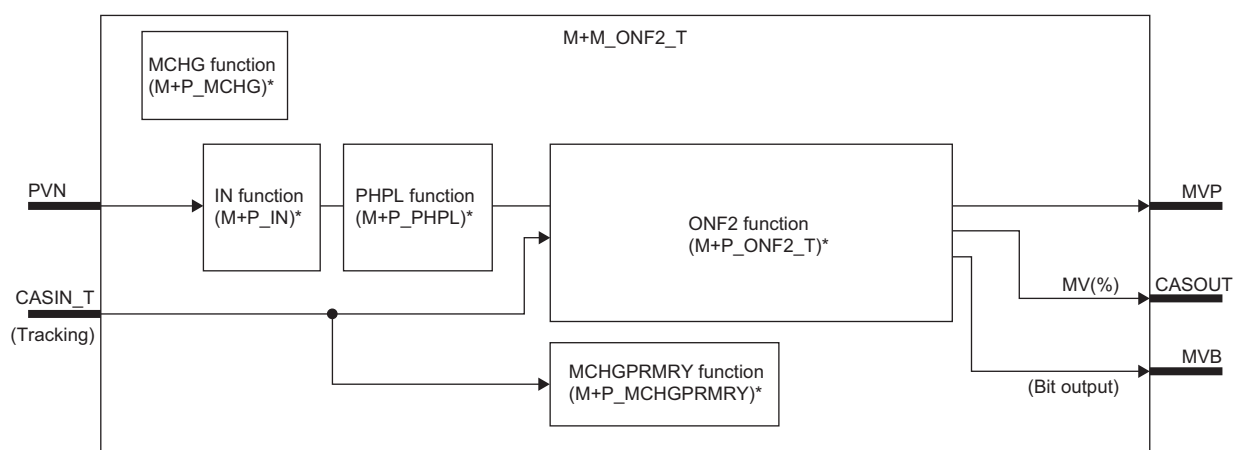
### Applicable tag type

ONF2

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN_T	Primary loop SV input (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVP	ΔMV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
MVB	ON/OFF output (turns on at MV ≥ 50%)	TRUE: ON FALSE: OFF	Output variable	BOOL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*5*6</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*2*3</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
ONF2_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
ONF2_TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
ONF2_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
ONF2_SVPTN_B1	Setting Value (SV) Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
MCHG_BNAL_MODE <sup>*2*3</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT
MCHGPRMRY_SEA_EN <sup>*2</sup>	When a sensor error occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_OOA_EN <sup>*2</sup>	When output open occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_SPA_EN <sup>*2</sup>	When a stop alarm occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_CASCASDR_EN <sup>*2</sup>	When the control mode is changed (CAS→AUT/MAN/CSV/CMV), switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*2 This public variable was added in version 1.070Y of the function block.

\*3 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*4 When TRUE (switch) is set, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*5 This public variable was added in version 1.085P of the function block.

\*6 If the version of the function block is lower than 1.085P, complies with the setting common to all tags.

## ■ Public variable (others)<sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■ Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

📖 Page 849 ONF2

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
ONF2 function	M+P_ONF2_T	Page 335 M+P_ONF2_T
MCHG function	M+P_MCHG	Page 378 M+P_MCHG
MCHGPRMRY function <sup>*1</sup>	M+P_MCHGPRMRY	Page 381 M+P_MCHGPRMRY

\*1 This function was added in version 1.070Y of the function block.

## Operation error

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

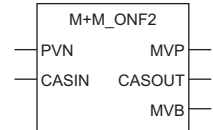


# 17.24 2 position ON/OFF Control (Disable Tracking for primary loop) (M+M\_ONF2)

## M+M\_ONF2

This FB performs two-position (on/off) control combining the functions of M+P\_IN, M+P\_PHPL, and M+P\_ONF2 into a single FB.

### FBD/LD



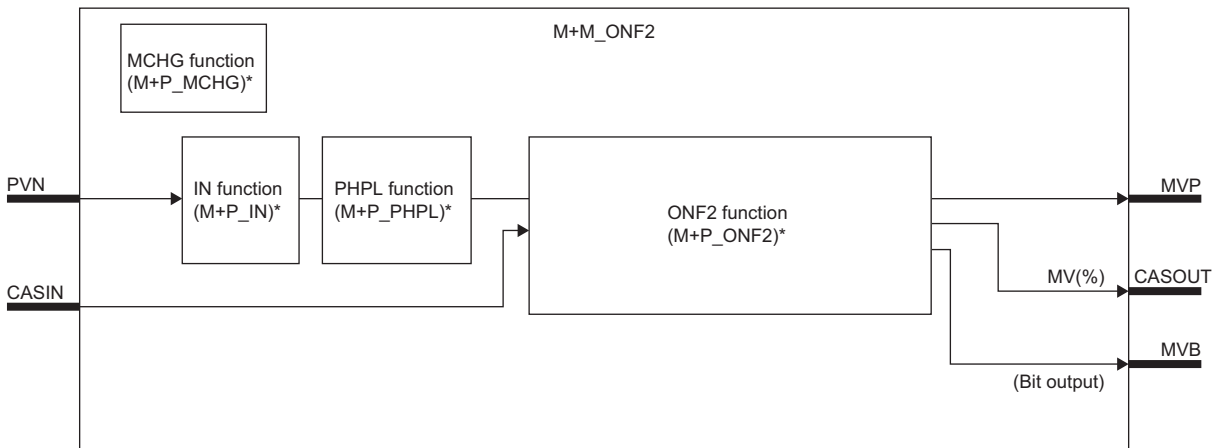
### Applicable tag type

ONF2

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN	Primary loop SV input (Unit: %)	0 to 100[%]	Input variable	REAL
MVP	Δ MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
MVB	ON/OFF output (turns on at MV ≥ 50%)	TRUE: ON FALSE: OFF	Output variable	BOOL

## ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*2</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*1</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
ONF2_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
ONF2_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
MCHG_BNAL_MODE <sup>*1</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT

\*1 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

\*2 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

## ■Public variable (others) <sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

📖 Page 849 ONF2

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
ONF2 function	M+P_ONF2	Page 338 M+P_ONF2
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

## Operation error

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

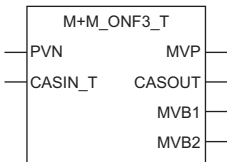
# 17.25 3 position ON/OFF Control (Enable Tracking for primary loop) (M+M\_ONF3\_T)

## M+M\_ONF3\_T

This FB performs three-position (on/off) control combining the functions of M+P\_IN, M+P\_PHPL, and M+P\_ONF3\_T into a single FB.

The primary loop can be tracked in the cascade connection. If a sensor error, output open, stop alarm, or control mode change occurs, the primary loop control mode is switched to MANUAL.

### FBD/LD



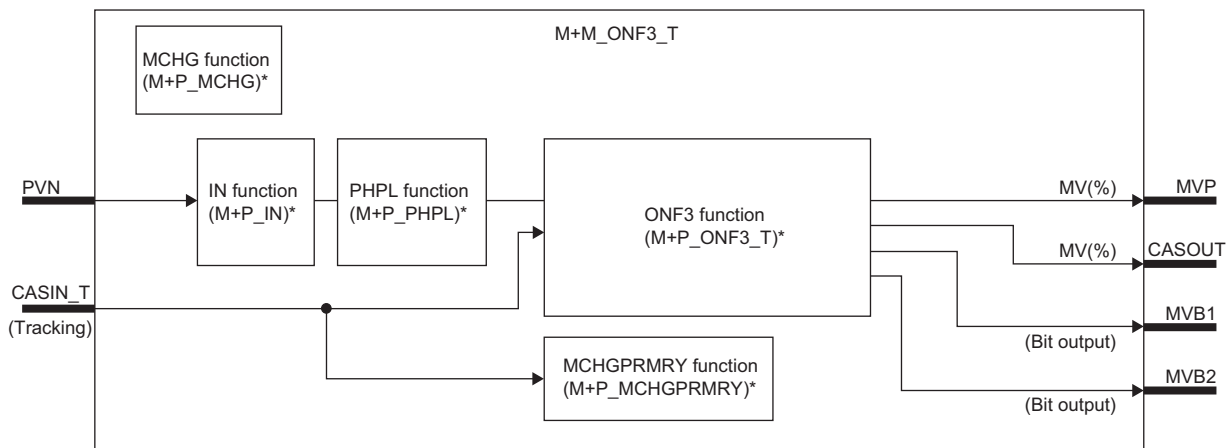
### Applicable tag type

ONF3

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN_T	Primary loop SV input (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVP	MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
MVB1	ON/OFF output (turns on at MV ≥ 75%)	TRUE: ON FALSE: OFF	Output variable	BOOL
MVB2	ON/OFF output (turns on at MV < 25%)	TRUE: ON FALSE: OFF	Output variable	BOOL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*5*</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*2*</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
ONF3_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
ONF3_TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
ONF3_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
ONF3_SVPTN_B1	Setting Value (SV) Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
MCHG_BNAL_MODE <sup>*2*</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT
MCHGPRMRY_SEA_EN <sup>*2</sup>	When a sensor error occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_OOA_EN <sup>*2</sup>	When output open occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_SPA_EN <sup>*2</sup>	When a stop alarm occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_CASCASDR_EN <sup>*2</sup>	When the control mode is changed (CAS→AUT/MAN/CSV/CMV), switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*2 This public variable was added in version 1.070Y of the function block.

\*3 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*4 When TRUE (switch) is set, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*5 This public variable was added in version 1.085P of the function block.

\*6 If the version of the function block is lower than 1.085P, complies with the setting common to all tags.

## ■ Public variable (others)<sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■ Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

📖 Page 852 ONF3

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
ONF3 function	M+P_ONF3_T	Page 341 M+P_ONF3_T
MCHG function	M+P_MCHG	Page 378 M+P_MCHG
MCHGPRMRY function <sup>*1</sup>	M+P_MCHGPRMRY	Page 381 M+P_MCHGPRMRY

\*1 This function was added in version 1.070Y of the function block.

## Operation error

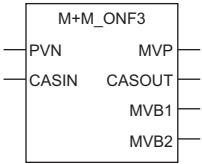
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.26 3 position ON/OFF Control (Disable Tracking for primary loop) (M+M\_ONF3)

## M+M\_ONF3

This FB performs three-position (on/off) control combining the functions of M+P\_IN, M+P\_PHPL, and M+P\_ONF3 into a single FB.

### FBD/LD



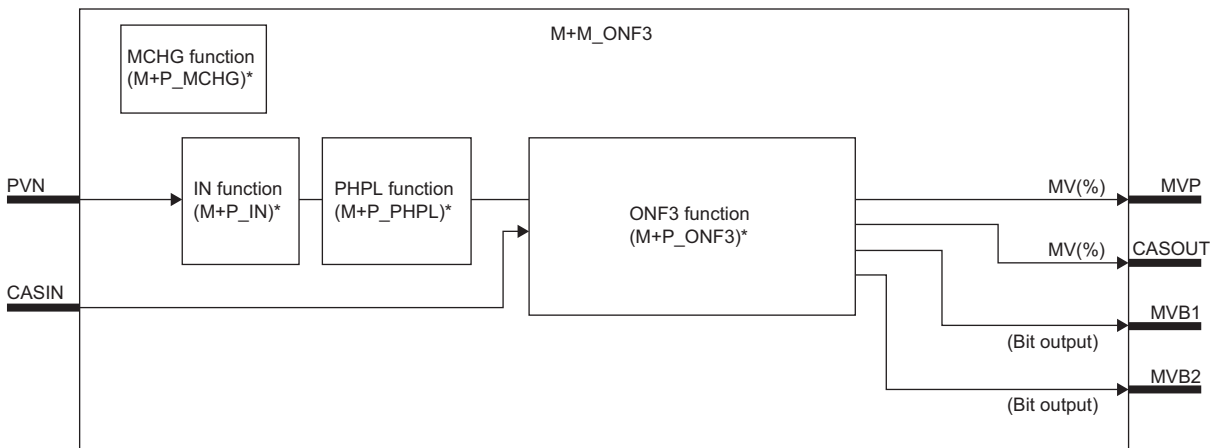
### Applicable tag type

ONF3

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN	Primary loop SV input (Unit: %)	0 to 100[%]	Input variable	REAL
MVP	ΔMV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
MVB1	ON/OFF output (turns on at MV ≥ 75%)	TRUE: ON FALSE: OFF	Output variable	BOOL
MVB2	ON/OFF output (turns on at MV < 25%)	TRUE: ON FALSE: OFF	Output variable	BOOL

## ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*2</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*1</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
ONF3_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0	User	INT
ONF3_SVPTN_BO	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
MCHG_BNAL_MODE <sup>*1</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT

\*1 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

\*2 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

## ■Public variable (others) <sup>\*1</sup>

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	NMIN to NMAX	0.0	User	REAL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

📄 Page 852 ONF3



## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
ONF3 function	M+P_ONF3	Page 345 M+P_ONF3
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

## Operation error

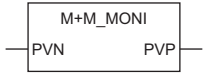
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.27 Monitor (M+M\_MONI)

## M+M\_MONI

This FB performs monitoring combining the functions of M+P\_IN and M+P\_PHPL into a single FB.

### FBD/LD



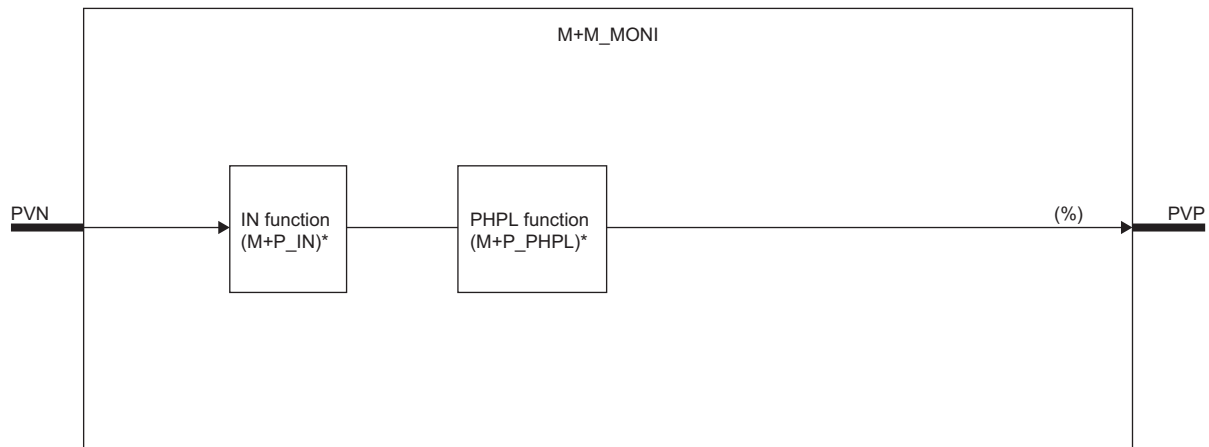
### Applicable tag type

MONI

### Control mode

MAN	AUT	CAS	CMV	CSV
—	—	—	—	—

### ■Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

## Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
PVP	PV output (Unit: %)	0 to 100[%]	Output variable	REAL

### ■Public variable (operation constant)


Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE*2	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE*1	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT

\*1 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

\*2 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

### ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

 Page 855 MONI

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL

## Operation error

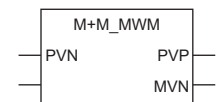
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.28 Manual Output with Monitor (M+M\_MWM)

## M+M\_MWM

This FB performs manual output with monitor combining the functions of M+P\_IN, M+P\_PHPL, and M+P\_MOUT into a single FB.

### FBD/LD



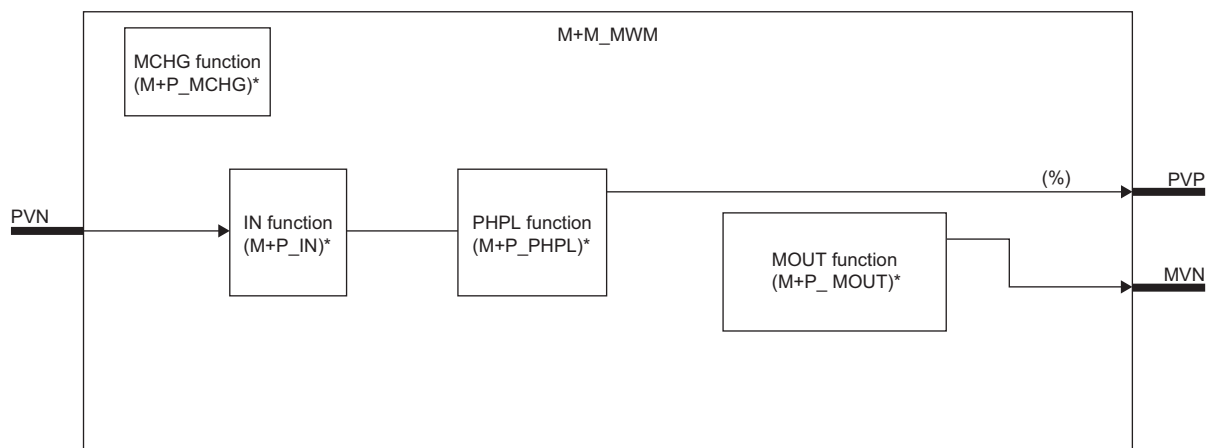
### Applicable tag type

MWM

### Control mode

MAN	AUT	CAS	CMV	CSV
○	—	—	○	—

### ■Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
PVP	PV output (Unit: %)	0 to 100[%]	Output variable	REAL
MVN	Output to a module	MOUT_NMIN to MOUT_NMAX	Output variable	REAL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*2</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*1</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
MOUT_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
MOUT_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
MCHG_BNAL_MODE <sup>*1</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT

\*1 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

\*2 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

## Public variable (others) <sup>\*1</sup>

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

☞ Page 860 MWM

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
MOUT function	M+P_MOUT	Page 206 M+P_MOUT
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

## Operation error

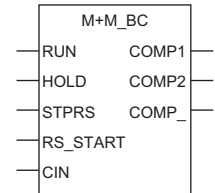
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.29 Batch Preparation (M+M\_BC)

## M+M\_BC

This FB performs batch preparation combining the functions of M+P\_PSUM and M+P\_BC into a single FB.

### FBD/LD



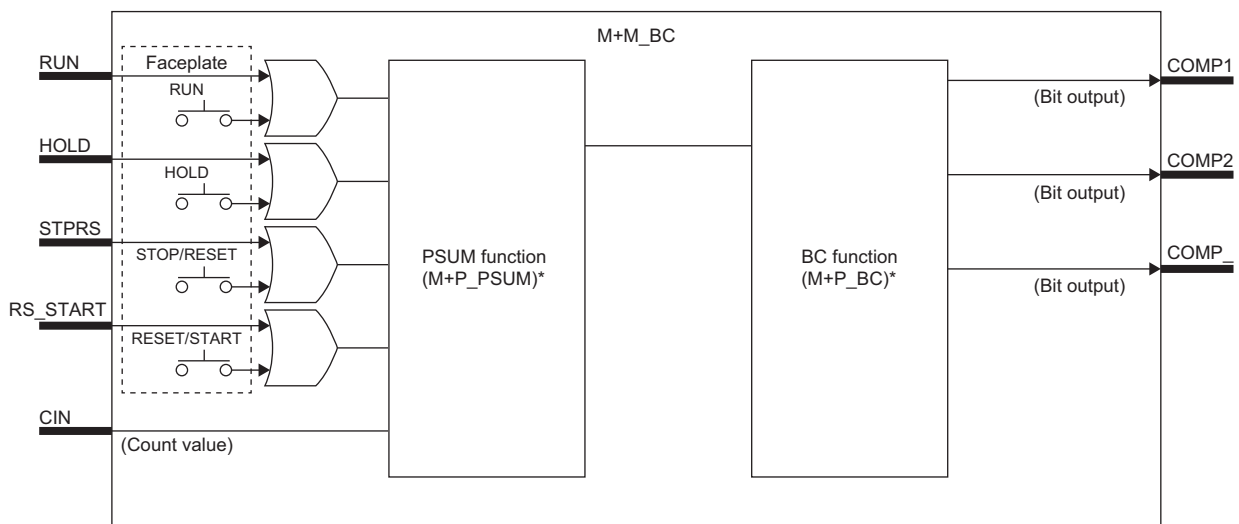
### Applicable tag type

BC

### Control mode

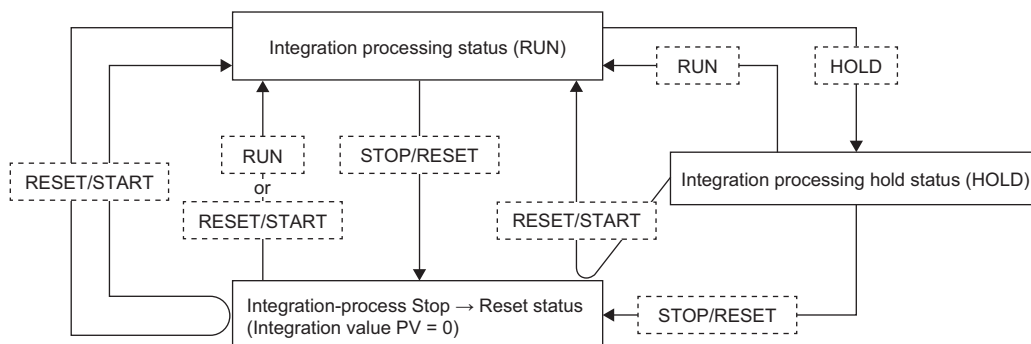
MAN	AUT	CAS	CMV	CSV
—	—	—	—	—

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

Indicates accumulation processing status according to input condition.



## Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
RUN	Integration start signal	TRUE: Execute FALSE: Stop	Input variable	BOOL
HOLD	Integration stop signal	TRUE: Execute FALSE: Stop	Input variable	BOOL
STPRS	Reset signal after integration stop	TRUE: Execute FALSE: Stop	Input variable	BOOL
RS_START	Start signal after integration reset	TRUE: Execute FALSE: Stop	Input variable	BOOL
CIN	Count value	Ring counter with the range of -2147483648 to 2147483647 (Pulse increment for each execution should be 32767 or less.)	Input variable	DINT
COMP1	Setting value 1 (SV1) completed output	TRUE: ON FALSE: OFF	Output variable	BOOL
COMP2	Setting value 2 (SV2) completed output	TRUE: ON FALSE: OFF	Output variable	BOOL
COMP_	Setting value (SV) completed output	TRUE: ON FALSE: OFF COMP_ will be TRUE when the count value (CIN) matches the setting value (SV).	Output variable	BOOL

### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
PSUM_W	Weight Per Pulse	1 to 999	1	User	INT
PSUM_U	Unit Conversion Constant	1, 10, 100, 1000	1	User	INT
PSUM_HILMT	High Limit Value of Integration	0 to 2147483647	2147483647	User	DINT
PSUM_SUMPTN	Integration Pattern	0: When an integrated value exceeds the integration high limit, the value is cleared to 0. 1: When an integrated value exceeds the integration high limit, the high limit value is held.	0	User	INT

### ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

☞ Page 863 BC

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
PSUM function	M+P_PSUM	Page 213 M+P_PSUM
BC function	M+P_BC	Page 216 M+P_BC

## Operation error

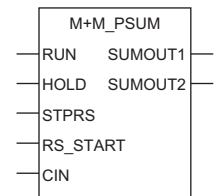
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.30 Pulse Integrator (M+M\_PSUM)

## M+M\_PSUM

This FB performs pulse integration combining the functions of M+P\_PSUM into a single FB.

### FBD/LD



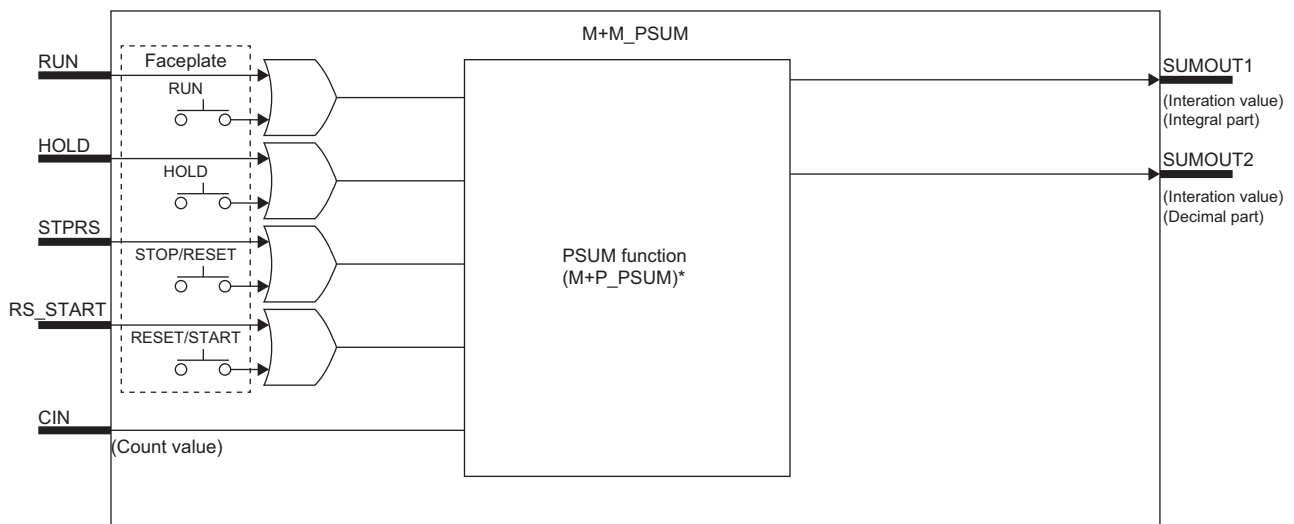
### Applicable tag type

PSUM

### Control mode

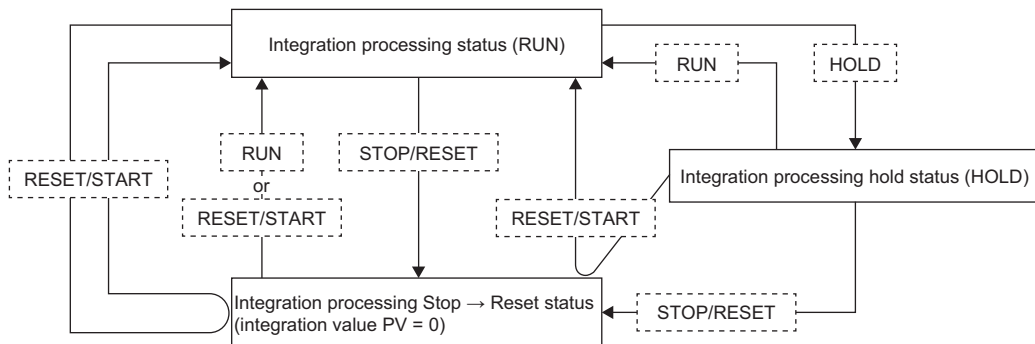
MAN	AUT	CAS	CMV	CSV
—	—	—	—	—

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

Indicates accumulation processing status according to input condition.





## Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
RUN	Integration start signal	TRUE: Execute FALSE: Stop	Input variable	BOOL
HOLD	Integration stop signal	TRUE: Execute FALSE: Stop	Input variable	BOOL
STPRS	Reset signal after integration stop	TRUE: Execute FALSE: Stop	Input variable	BOOL
RS_START	Start signal after integration reset	TRUE: Execute FALSE: Stop	Input variable	BOOL
CIN	Count value	Ring counter with the range of -2147483648 to 2147483647 (Pulse increment for each execution should be 32767 or less.)	Input variable	DINT
SUMOUT1	Integration value (integral part) output	0 to 99999999	Output variable	DINT
SUMOUT2	Integration value (decimal part) output	0 to 999	Output variable	DINT

### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
PSUM_W	Weight Per Pulse	1 to 999	1	User	INT
PSUM_U	Unit Conversion Constant	1, 10, 100, 1000	1	User	INT
PSUM_HILMT	High Limit Value of Integration	0 to 2147483647	2147483647	User	DINT
PSUM_SUMPTN	Integration Pattern	0: When an integrated value exceeds the integration high limit, the value is cleared to 0. 1: When an integrated value exceeds the integration high limit, the high limit value is held.	0	User	INT

### ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

📖 Page 865 PSUM

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
PSUM function	M+P_PSUM	Page 213 M+P_PSUM

## Operation error

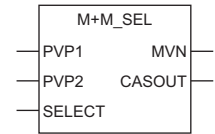
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.31 Loop Selector (Disable Tracking for primary loop) (M+M\_SEL)

## M+M\_SEL

This FB performs loop selector combining the functions of M+P\_SEL into a single FB.

### FBD/LD



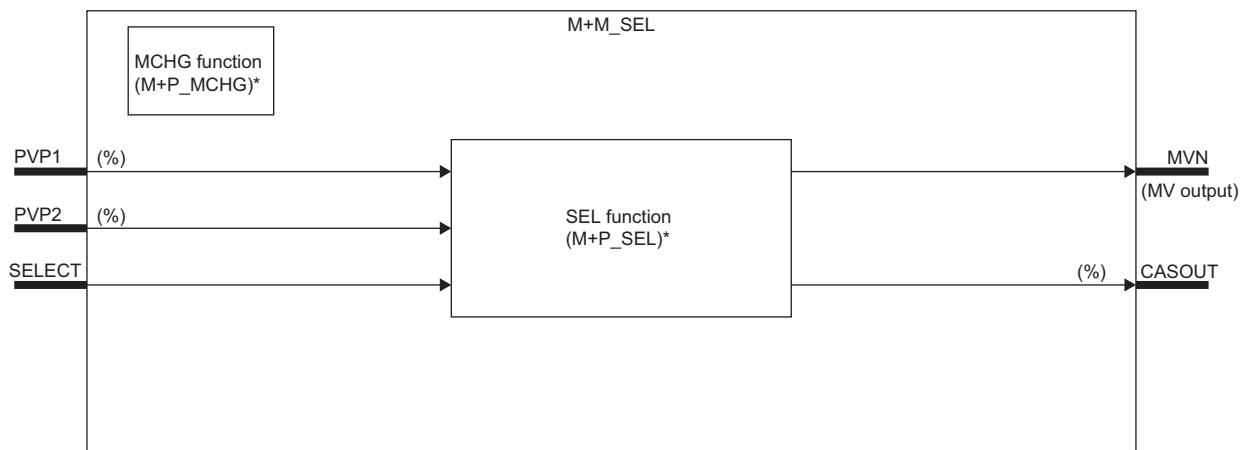
### Applicable tag type

SEL

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP1	PV input (Unit: %)	0 to 100[%]	Input variable	REAL
PVP2	PV input (Unit: %)	0 to 100[%]	Input variable	REAL
SELECT	Selection signal	TRUE: PVP2 FALSE: PVP1	Input variable	BOOL
MVN	Output to a module	SEL_NMIN to SEL_NMAX	Output variable	REAL
CASOUT	Cascade output (Unit: %)	0 to 100[%]	Output variable	REAL

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
SEL_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
SEL_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
MCHG_BNAL_MODE <sup>*1</sup>	Processing selection when output open occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT

\*1 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

## Public variable (others) <sup>\*1</sup>

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

 Page 866 SEL

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
SEL function	M+P_SEL	Page 363 M+P_SEL
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

## Operation error

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

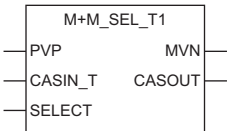
# 17.32 Loop Selector (Enable Tracking for primary loop only by CASIN\_T) (M+M\_SEL\_T1)

## M+M\_SEL\_T1

This FB performs loop selector combining the functions of M+P\_SEL\_T1 into a single FB.

The input 1 is not tracked in the cascade connection. The primary loop of the input 2 is tracked, but tracking from the secondary loop is not performed. If output open, stop alarm, or control mode change occurs, the control mode of the primary loop of the input 2 is switched to MANUAL.

### FBD/LD



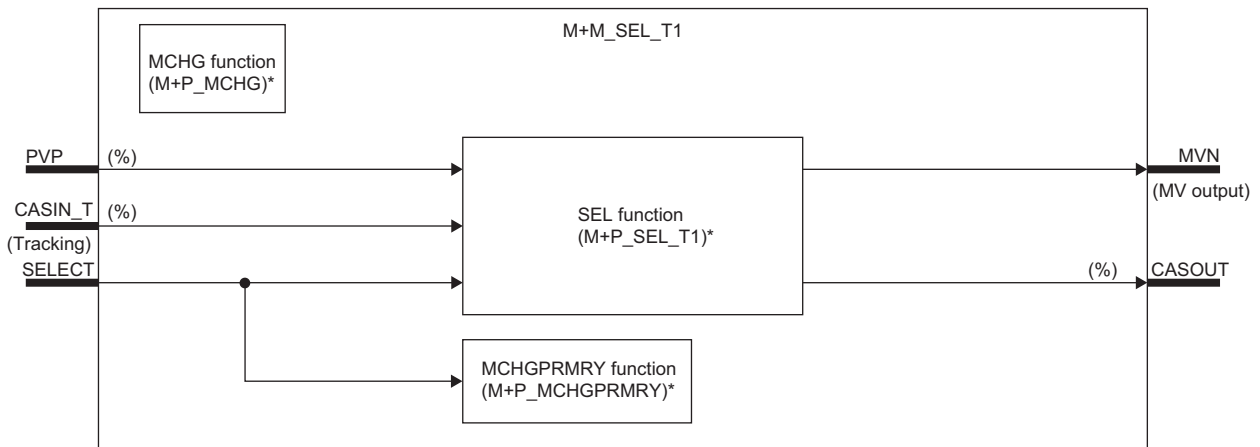
### Applicable tag type

SEL

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVP	PV input (Unit: %)	0 to 100[%]	Input variable	REAL
CASIN_T	PV input (Unit: %) (With tracking) (Indirect address)* <sup>1</sup>	0 to 100[%]	Input variable	DWORD
SELECT	Selection signal	TRUE: CASIN_T FALSE: PVP	Input variable	BOOL
MVN	Output to a module	SEL_NMIN to SEL_NMAX	Output variable	REAL
CASOUT	Cascade output (Unit: %)	0 to 100[%]	Output variable	REAL

\*1 Connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
SEL_NMAX	Output Conversion High Limit	-999999 to 999999	0.0	User	REAL
SEL_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
SEL_TRK	Tracking Flag	0: Not executed 1: Executed	0	User	INT
SEL_SVPTN_B4	CASIN_T Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
MCHG_BNAL_MODE <sup>*1*2</sup>	Processing selection when output open occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT
MCHGPRMRY_OOA_EN <sup>*1</sup>	When output open occurs, switches the primary loop to MANUAL mode <sup>*3</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_SPA_EN <sup>*1</sup>	When a stop alarm occurs, switches the primary loop to MANUAL mode <sup>*3</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_CASCASDR_EN <sup>*1</sup>	When the control mode is changed (CAS→AUT/MAN/CSV/CMV), switches the primary loop to MANUAL mode <sup>*3</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL

\*1 This public variable was added in version 1.070Y of the function block.

\*2 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*3 Switches the primary loop of the input 2 (CASIN\_T) to MANUAL, regardless of the selection of the input variable SELECT.

## Public variable (others) <sup>\*1</sup>

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

 Page 866 SEL

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
SEL function	M+P_SEL_T1	Page 366 M+P_SEL_T1
MCHG function	M+P_MCHG	Page 378 M+P_MCHG
MCHGPRMRY function <sup>*1</sup>	M+P_MCHGPRMRY	Page 381 M+P_MCHGPRMRY

\*1 This function was added in version 1.070Y of the function block.

## Operation error

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

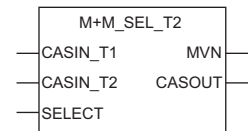
# 17.33 Loop Selector (Enable Tracking for primary loop by CASIN\_T1/\_T2) (M+M\_SEL\_T2)

## M+M\_SEL\_T2

This FB performs loop selector combining the functions of M+P\_SEL\_T2 into a single FB.

In the cascade connection, the primary loop of the inputs 1 and 2 is tracked, but tracking from the secondary loop is not performed. If output open, stop alarm, or control mode change occurs, the control mode of the primary loop of the inputs 1 and 2 is switched to MANUAL.

### FBD/LD



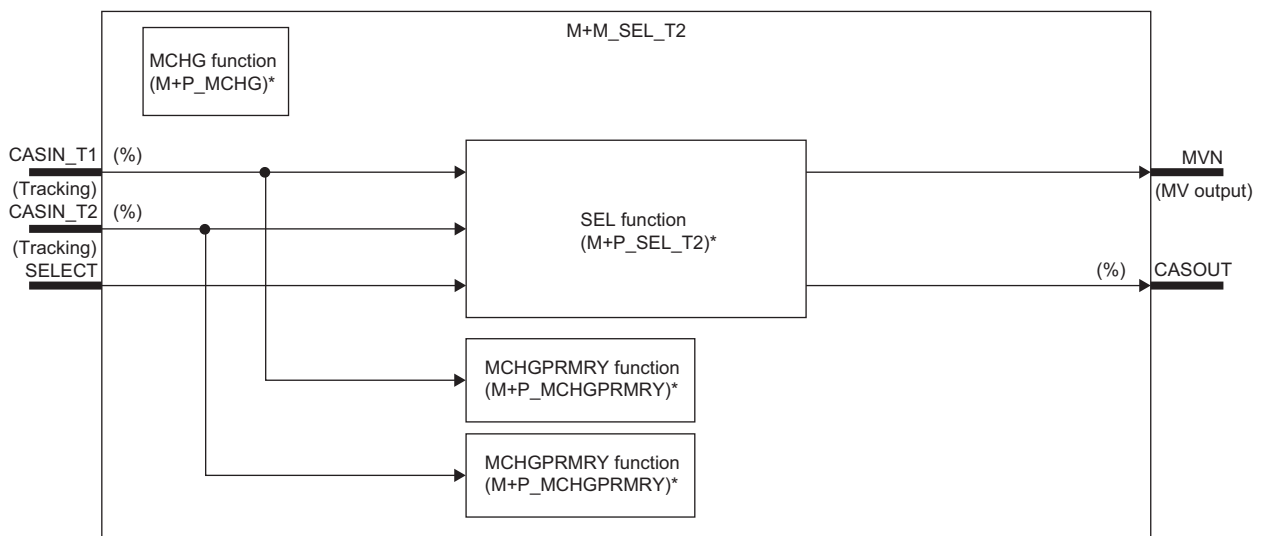
### Applicable tag type

SEL

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
CASIN_T1	PV input (Unit: %) (With tracking) (Indirect address)* <sup>1</sup>	0 to 100[%]	Input variable	DWORD
CASIN_T2	PV input (Unit: %) (With tracking) (Indirect address)* <sup>1</sup>	0 to 100[%]	Input variable	DWORD
SELECT	Selection signal	TRUE: CASIN_T2 FALSE: CASIN_T1	Input variable	BOOL
MVN	Output to a module	SEL_NMIN to SEL_NMAX	Output variable	REAL
CASOUT	Cascade output (Unit: %)	0 to 100[%]	Output variable	REAL

\*<sup>1</sup> Connect CASOUT\_T of the primary loop with the input variable CASIN\_T1 and CASIN\_T2.

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
SEL_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
SEL_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
SEL_TRK	Tracking Flag	0: Not executed 1: Executed	0	User	INT
SEL_SVPTN_B1	CASIN_T1 Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SEL_SVPTN_B2	CASIN_T2 Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SEL_SVPTN_B3	CASIN_T1 Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
SEL_SVPTN_B4	CASIN_T2 Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
MCHG_BNAL_MODE <sup>*1*</sup>	Processing selection when output open occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT
MCHGPRMRY_OOA_EN <sup>*1</sup>	When output open occurs, switches the primary loop to MANUAL mode <sup>*3</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_SPA_EN <sup>*1</sup>	When a stop alarm occurs, switches the primary loop to MANUAL mode <sup>*3</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_CASCASDR_EN <sup>*1</sup>	When the control mode is changed (CAS→AUT/MAN/CSV/CMV), switches the primary loop to MANUAL mode <sup>*3</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL

\*1 This public variable was added in version 1.070Y of the function block.

\*2 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*3 Switches the primary loop of the input 1 (CASIN\_T1) and the input 2 (CASIN\_T2) to MANUAL, regardless of the selection of the input variable SELECT.

## Public variable (others) <sup>\*1</sup>

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

 Page 866 SEL

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
SEL function	M+P_SEL_T2	Page 370 M+P_SEL_T2
MCHG function	M+P_MCHG	Page 378 M+P_MCHG
MCHGPRMRY function <sup>*1</sup>	M+P_MCHGPRMRY	Page 381 M+P_MCHGPRMRY

\*1 This function was added in version 1.070Y of the function block.

## Operation error

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.



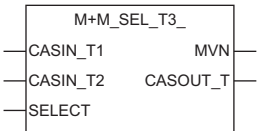
# 17.34 Loop Selector (Enable Tracking from secondary loop to primary loop) (M+M\_SEL\_T3\_)

## M+M\_SEL\_T3\_

This FB performs loop selector combining the functions of M+P\_SEL\_T3\_ into a single FB.

In the cascade connection, performs loop selector combining the functions of M+P\_SEL\_T3\_ into a single FB. The primary loop of the input 1 and 2 can be tracked from the secondary loop. If output open, stop alarm, or control mode change occurs, the control mode of the primary loop of the inputs 1 and 2 is switched to MANUAL.

### FBD/LD



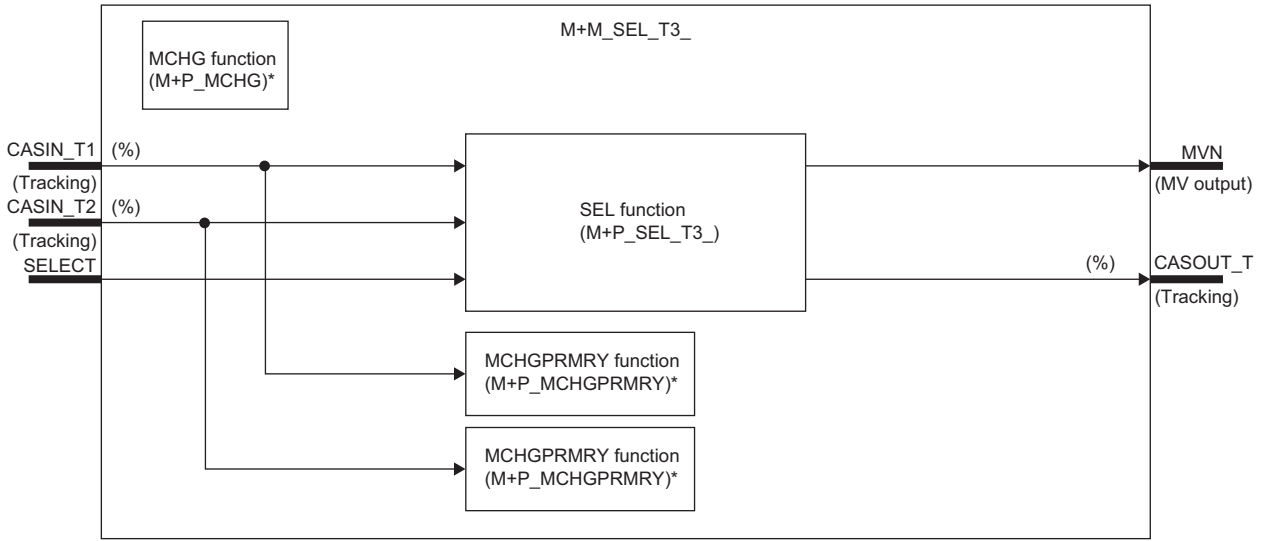
### Applicable tag type

SEL

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
CASIN_T1	PV input (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
CASIN_T2	PV input (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
SELECT	Selection signal	TRUE: CASIN_T2 FALSE: CASIN_T1	Input variable	BOOL
MVN	Output to a module	SEL_NMIN to SEL_NMAX	Output variable	REAL
CASOUT_T	Cascade output (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

## Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
SEL_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
SEL_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
SEL_TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
SEL_SVPTN_B1	CASIN_T1 Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SEL_SVPTN_B2	CASIN_T2 Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SEL_SVPTN_B3	CASIN_T1 Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
SEL_SVPTN_B4	CASIN_T2 Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
SEL_SVPTN_B5	Tracking to Non-selected loop	TRUE: Execute FALSE: Not execute	FALSE	User	BOOL
MCHG_BNAL_MODE <sup>*2*3</sup>	Processing selection when output open occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT
MCHGPRMRY_OOA_EN <sup>*2</sup>	When output open occurs, switches the primary loop to MANUAL mode <sup>*4*5</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_SPA_EN <sup>*2</sup>	When a stop alarm occurs, switches the primary loop to MANUAL mode <sup>*4*5</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_CASCASDR_EN <sup>*2</sup>	When the control mode is changed (CAS→AUT/MAN/CSV/CMV), switches the primary loop to MANUAL mode <sup>*4*5</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T1 and CASIN\_T2.

\*2 This public variable was added in version 1.070Y of the function block.

\*3 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*4 When TRUE (switch) is set, connect CASOUT\_T of the primary loop with each of the input variables CASIN\_T1 and CASIN\_T2.

\*5 Switches the primary loop of the input 1 (CASIN\_T1) and the input 2 (CASIN\_T2) to MANUAL, regardless of the selection of the input variable SELECT.

### Point

When the mode switching processing from the secondary loop is also reflected on the primary loop, set MCHGPRMRY\_CASCASDR\_EN to TRUE (switch) when changing the control mode.

## Public variable (others)<sup>\*1</sup>

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

☞ Page 866 SEL

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
SEL function	M+P_SEL_T3_	Page 374 M+P_SEL_T3_
MCHG function	M+P_MCHG	Page 378 M+P_MCHG
MCHGPRMRY function <sup>*1</sup>	M+P_MCHGPRMRY	Page 381 M+P_MCHGPRMRY

\*1 This function was added in version 1.070Y of the function block.

## Operation error

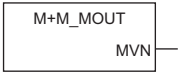
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.35 Manual Output (M+M\_MOUT)

## M+M\_MOUT

This FB performs manual output combining the functions of M+P\_MOUT into a single FB.

### FBD/LD



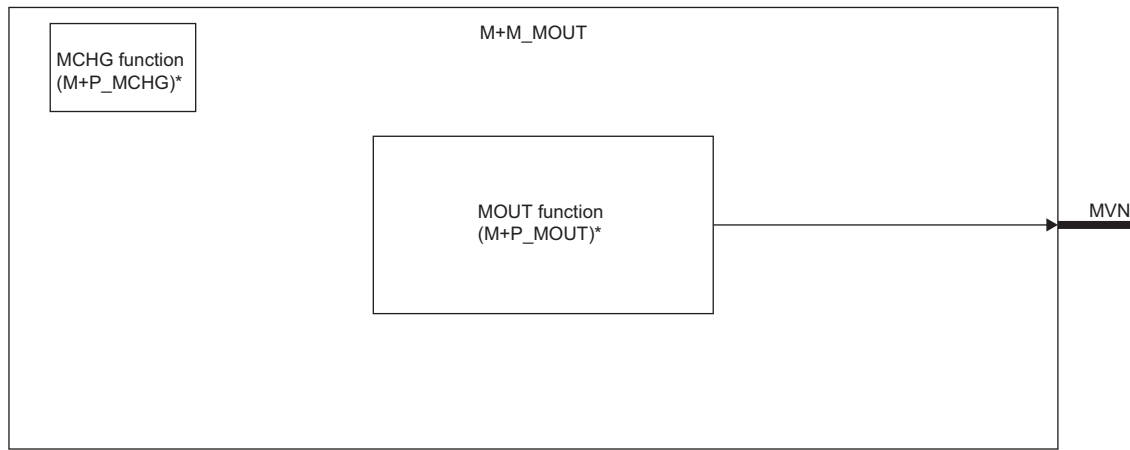
### Applicable tag type

MOUT

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	-	-	<input type="radio"/>	—

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
MVN	Output to a module	MOUT_NMIN to MOUT_NMAX	Output variable	REAL

#### Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MOUT_NMAX	Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
MOUT_NMIN	Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
MCHG_BNAL_MODE <sup>*1</sup>	Processing selection when output open occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT

\*1 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

**Public variable (others) \*1**

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

**Tag data**

For details on tag data that is read or written with this tag FB, refer to the following.

☞ Page 868 MOUT

**Processing details**

This tag FB consists of the following function blocks.

Item	Function block	Reference
MOUT function	M+P_MOUT	Page 206 M+P_MOUT
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

**Operation error**

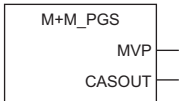
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.36 Program Setter (M+M\_PGS)

## M+M\_PGS

This FB performs program settings combining the functions of M+P\_PGS into a single FB.

### FBD/LD



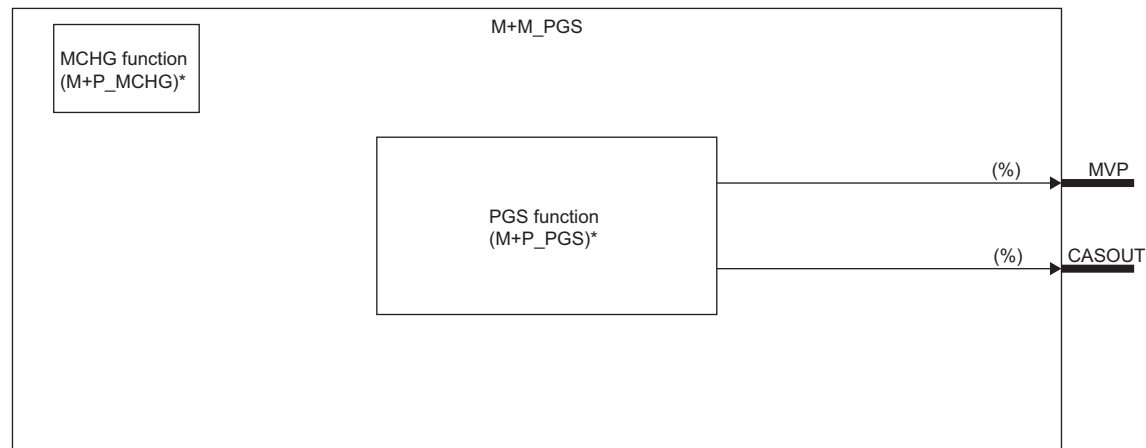
### Applicable tag type

PGS

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
MVP	MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT	Cascade output (Unit: %)	0 to 100[%]	Output variable	REAL

#### Public variable (others) \*1

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■ Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

☞ Page 869 PGS

### Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
PGS function	M+P_PGS	Page 348 M+P_PGS
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

### Operation error

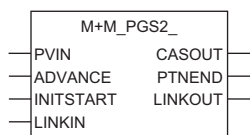
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

# 17.37 Multi-Point Program Setter (M+M\_PGS2\_)

## M+M\_PGS2\_

This FB performs program settings combining the functions of M+P\_PGS2\_ into a single FB. Multiple FBs can be connected to execute time width sets and setting value programs of 32 steps or more.

### FBD/LD



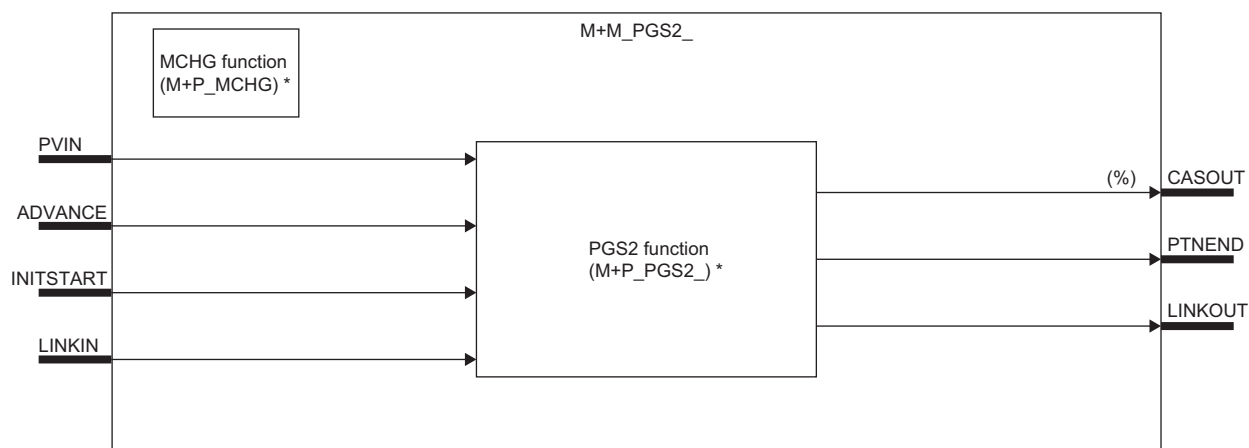
### Applicable tag type

PGS2

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	—	—	—

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVIN	Process input (Engineering value)	-32768 to 32767	Input variable	REAL
ADVANCE	Advance command	TRUE, FALSE	Input variable	BOOL
INITSTART	Initial start command	TRUE, FALSE	Input variable	BOOL
LINKIN	Link input (Indirect address)	—	Input variable	DWORD
CASOUT	Cascade output (Unit: %)	0 to 100[%]	Output variable	REAL
PTNEND	Pattern end output	TRUE, FALSE	Output variable	BOOL
LINKOUT	Link output (Indirect address)	—	Output variable	DWORD



## ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
PGS2_PVSTARTNO	PV Start Search Start Step	1 to 32	1	User	INT
PGS2_PVENDNO	PV Start Search End Step	1 to 32	32	User	INT
PGS2_PRIMARY	Lead FB specified	TRUE: Lead FALSE: Following	TRUE	User	BOOL

## ■Public variable (others) \*1

- M+P\_PGS2\_ processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
PGS2_TCNT	Second counter for minute mode.	0 to 59	0	System	INT
PGS2_TMCNT	Millisecond counter for second mode.	0 to 999	0	System	INT

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1, 2 1: MAN 2: AUT	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

☞ Page 872 PGS2

### Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
PGS2 function	M+P_PGS2_	Page 352 M+P_PGS2_
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

### Operation error

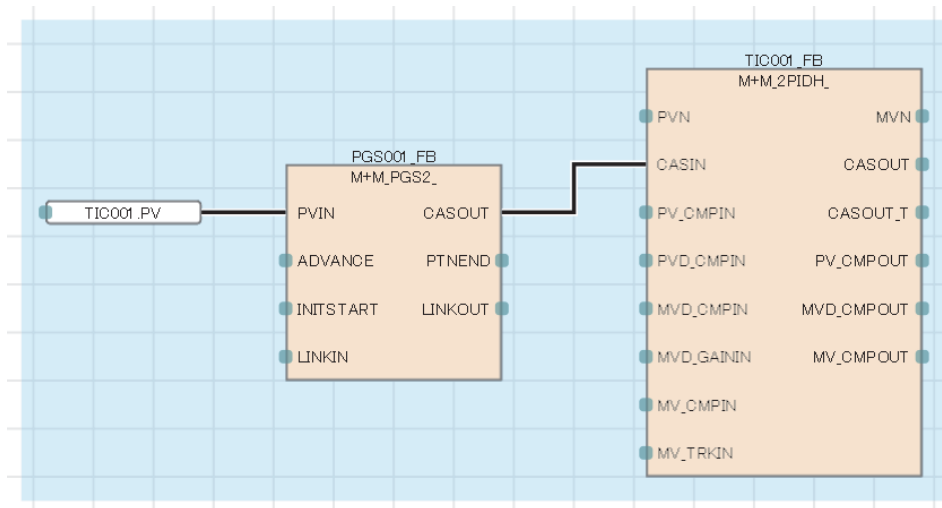
An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

## Program example

### ■ When the output of a single M+M\_PGS2\_ function block is used

Set the following items.

Type/pin	Variable name	Description	Setting/connection method
Public variable	PGS2_PRIMARY	Lead FB specified	TRUE
Input pin	LINKIN	Link input	Not connected.



**■When multiple M+M\_PGS2\_ function blocks are connected (When a program with 32 steps or more is used)**

Set the following items.

Target FB	Type/pin	Variable name	Description	Setting/connection method
Lead FB	Public variable	PGS2_PRIMARY	Lead FB specified	TRUE
	Input pin	LINKIN	Link input	Not connected.
	Output pin	PTNEND	Pattern end output	Connected with INITSTART of the following FB.
		LINKOUT	Link output	Connected with LINKIN of the following FB.
Following FB	Public variable	PGS2_PRIMARY	Lead FB specified	FALSE
	Input pin	INITSTART	Initial start command	Connected with PTNEND of the preceding FB.
		LINKIN	Link input	Connected with LINKOUT of the preceding FB.
	Output pin	PTNEND	Pattern end output	Connected with INITSTART of the following FB.
		LINKOUT	Link output	Connected with LINKIN of the following FB.
	Last FB	Public variable	PGS2_PRIMARY	Lead FB specified
Input pin		INITSTART	Initial start command	Connected with PTNEND of the preceding FB.
		LINKIN	Link input	Connected with LINKOUT of the preceding FB.
Output pin		CASOUT	Cascade output (Unit: %)	Connected with CASIN of the following tag FB such as PID.

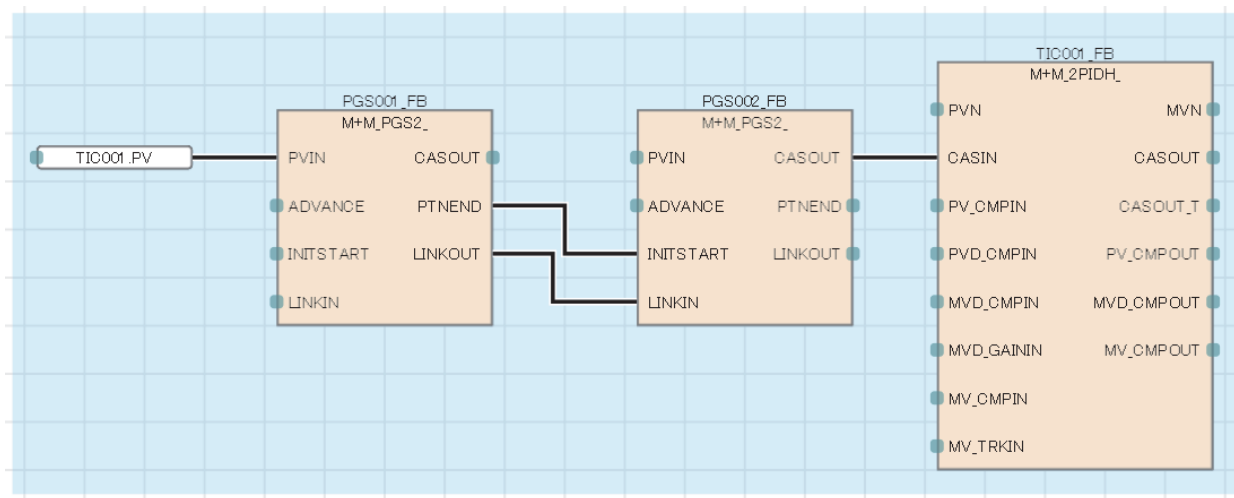
• Operation description

PGS001 (first program setter FB) is started in the AUT mode, and the output pin PTNEND turns on for one cycle at completion.

When PGS002 (second program setter FB) receives the PTNEND output of PGS001, the mode is changed to the AUT mode by the INITSTART command and the control is switched.

For details, refer to the following.

☞ Page 352 M+P\_PGS2\_

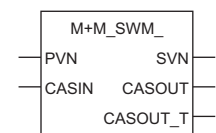


# 17.38 Manual Setter with Monitor (M+M\_SWM\_)

## M+M\_SWM\_

This FB performs manual settings with monitor combining the functions of M+P\_IN, M+P\_PHPL, and M+P\_MSET\_ into a single FB.

### FBD/LD



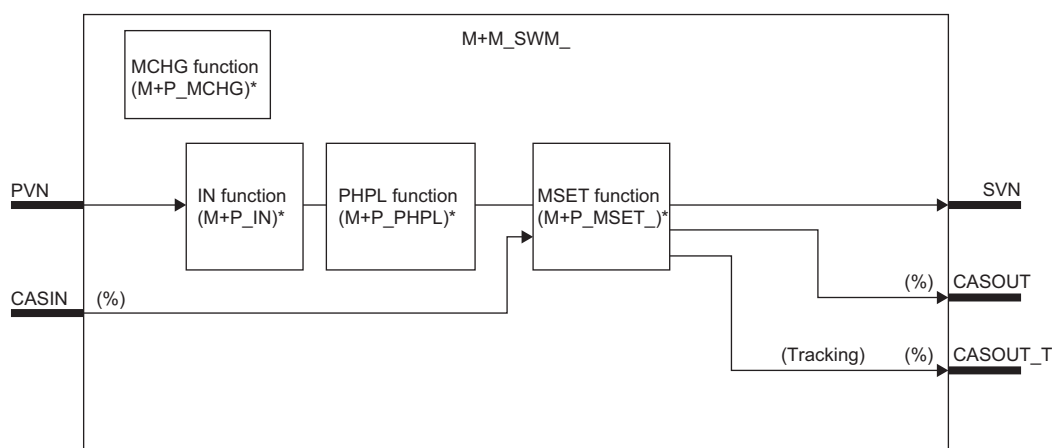
### Applicable tag type

SWM

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.

### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Input from a module	-999999 to 999999	Input variable	REAL
CASIN	Cascade SV input (unit: %)	0 to 100[%]	Input variable	REAL
SVN	Output to a module	-999999 to 999999	Output variable	REAL
CASOUT	Cascade MV output (Unit: %)	0 to 100[%]	Output variable	REAL
CASOUT_T	Cascade SV output (unit: %) (With tracking) (Indirect address)	0 to 100[%]	Output variable	DWORD

## ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*3</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*2</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
MSET_SVPTN_B0	Setting Value (SV) Used <sup>*1</sup>	TRUE: Not used FALSE: Used	TRUE	User	BOOL
MSET_SVLM_TEN	SV High/Low Limiter	TRUE: Execute FALSE: Not execute	FALSE	User	BOOL
MSET_DVLS	Large Deviation Alarm Hysteresis	0 to 100	0 to 100	User	REAL
MSET_PN	Reverse Action/Direct Action	0: Reverse action 1: Direct action	0 to 1	User	INT
MCHG_BNAL_MODE <sup>*2</sup>	Processing selection when sensor error/open output occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT

\*1 When SVPTN\_B0 is TRUE, CASIN inputs cannot be used even though the mode is changed to the CAS mode.

\*2 This public variable was added in version 1.070Y of the function block. If the version is lower than 1.070Y, complies with the setting common to all tags.

\*3 This public variable was added in version 1.085P of the function block. If the version is lower than 1.085P, complies with the setting common to all tags.

## ■Public variable (others) <sup>\*1</sup>

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 3, 5 1: MAN 2: AUT 3: CAS 5: CSV	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

📄 Page 857 SWM

## Processing details

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function	M+P_PHPL	Page 331 M+P_PHPL
MSET function	M+P_MSET_	Page 219 M+P_MSET_
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

## Operation error

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

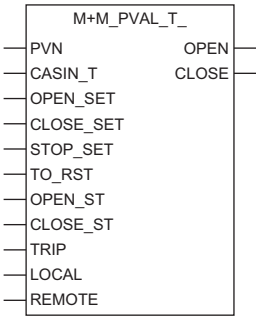
# 17.39 Position Proportional Output (M+M\_PVAL\_T\_)

## M+M\_PVAL\_T\_

This FB outputs open/close command bits to operate the motor valve opening according to the deviation of opening between the feedback input and setting value.

The primary loop can be tracked in the cascade connection. If a sensor error, stop alarm, tag stop, or control mode change occurs, the primary loop control mode is switched to MANUAL.

### FBD/LD



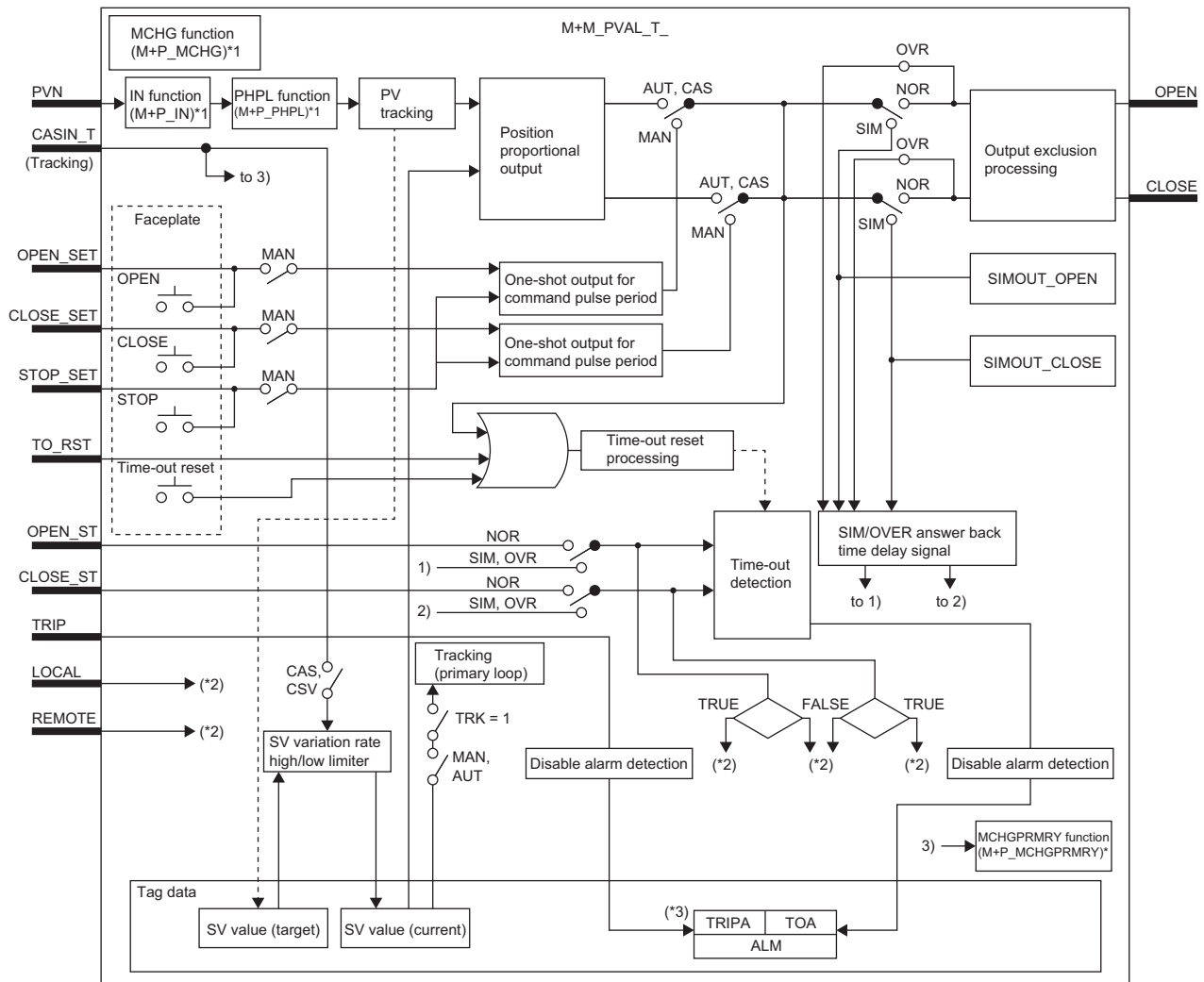
### Applicable tag type

PVAL

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	—	<input type="radio"/>

## Block diagram



\*1 (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.  
 \*2 For displaying the faceplate status.  
 \*3 Indicates bit item.

## Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
PVN	Valve opening feedback input	-999999 to 999999	Input variable	REAL
CASIN_T	Primary loop SV input (Unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
OPEN_SET	External input of OPEN operation	TRUE, FALSE FALSE → TRUE: OPEN	Input variable	BOOL
CLOSE_SET	External input of CLOSE operation	TRUE, FALSE FALSE → TRUE: CLOSE	Input variable	BOOL
STOP_SET	External input of STOP operation	TRUE, FALSE (FALSE → TRUE: Sets OPEN and CLOSE to FALSE)	Input variable	BOOL
TO_RST	Time-out error external reset input	TRUE, FALSE FALSE → TRUE: Time-out reset	Input variable	BOOL
OPEN_ST	Open status answer input	TRUE: During OPEN output FALSE: —	Input variable	BOOL
CLOSE_ST	Close status answer input	TRUE: During CLOSE output FALSE: —	Input variable	BOOL
TRIP	External failure (TRIP) input	TRUE: Detected FALSE: Reset	Input variable	BOOL



Variable name	Description	Recommended range	Type	Data type
LOCAL	Local operation selection signal	TRUE: Valid FALSE: Invalid	Input variable	BOOL
REMOTE	Remote operation selection signal	TRUE: Valid FALSE: Invalid	Input variable	BOOL
OPEN	Open command signal (OPEN) ON output	TRUE: Run FALSE: —	Output variable	BOOL
CLOSE	Close command signal (CLOSE) ON output	TRUE: Run FALSE: —	Output variable	BOOL

### ■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
IN_NMAX	Input High Limit	-999999 to 999999	100.0	User	REAL
IN_NMIN	Input Low Limit	-999999 to 999999	0.0	User	REAL
IN_HH	High Limit Range Error	-999999 to 999999	102.0	User	REAL
IN_H	High Limit Range Error Reset	-999999 to 999999	100.0	User	REAL
IN_L	Low Limit Range Error Reset	-999999 to 999999	0.0	User	REAL
IN_LL	Low Limit Range Error	-999999 to 999999	-2.0	User	REAL
IN_LMT_TYPE <sup>*5*6</sup>	Input limiter processing selection	0: Comply with the setting common to all tags 1: Perform limiter processing 2: Do not perform limiter processing	0	User	INT
IN_SEA_OTYPE <sup>*2*3</sup>	Hold processing selection for input function when sensor error occurs	0: Comply with the setting common to all tags 1: Hold 2: Continue operation	0	User	INT
DVLS	Large Deviation Alarm Hysteresis	0 to 100	2.0	User	REAL
PVTRK_EN	PV Tracking Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
LMTOUT_EN	Output at High or Low Limit of Valve Opening Execution condition	TRUE: Output when PV is 0% or 100% FALSE: Not output when PV is 0% or 100%	FALSE	User	BOOL
TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
SVPTN_B0	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SVPTN_B1	Setting Value (SV) Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
SVLMT_EN	SV High/Low Limiter	TRUE: Execute FALSE: Not execute	FALSE	User	BOOL
MCHG_BNAL_MODE <sup>*2*3</sup>	Processing selection when sensor error occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT
MCHGPRMRY_SEA_EN <sup>*2</sup>	When a sensor error occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_SPA_EN <sup>*2</sup>	When a stop alarm occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_TSTP_EN <sup>*2</sup>	When tag stop occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_CASCASDR_EN <sup>*2</sup>	When the control mode is changed (CAS→AUT/MAN/CSV), switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*2 This public variable was added in version 1.070Y of the function block.

\*3 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*4 When TRUE (switch) is set, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*5 This public variable was added in version 1.085P of the function block.

\*6 If the version of the function block is lower than 1.085P, complies with the setting common to all tags.

## ■Public variable (others) \*1

- Simulation processing

Variable name	Description	Recommended range	Initial value	Set by	Data type
SIMIN	Simulation Input	0 to 100	0.0	User	REAL
SIMOUT_OPEN	Simulation Output of Open Command Signal	TRUE, FALSE	FALSE	System	BOOL
SIMOUT_CLOSE	Simulation Output of Close Command Signal	TRUE, FALSE	FALSE	System	BOOL

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 3, 5 1: MAN 2: AUT 3: CAS 5: CSV	1	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

☞ Page 875 PVAL

## ■Function of the components

This tag FB consists of the following function blocks.

Item	Function block	Reference
IN function	M+P_IN	Page 184 M+P_IN
PHPL function*1	M+P_PHPL	Page 331 M+P_PHPL
MCHG function	M+P_MCHG	Page 378 M+P_MCHG
MCHGPRMRY function*2	M+P_MCHGPRMRY	Page 381 M+P_MCHGPRMRY

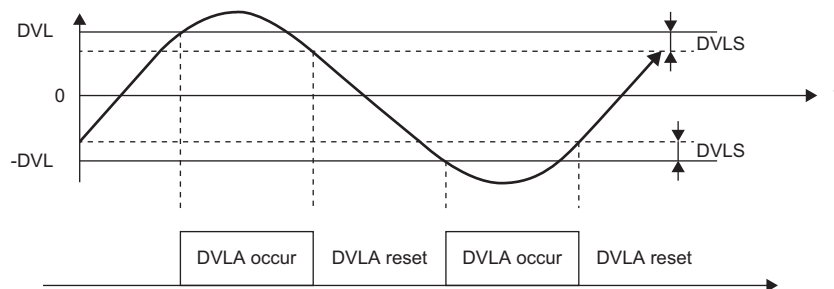
\*1 The high/low limit of engineering values are fixed to 100% and 0%.

\*2 This function was added in version 1.070Y of the function block.

## Processing details

### ■Deviation check

- This function block performs deviation check processing.



Condition	Alarm (ALM)
	Large deviation (DVLA)
$DVL <  DV $	TRUE (Detected)
$ DV  \leq (DVL - DVLS)$	FALSE (Reset)

DV: Deviation (%), DVLS: Large deviation alarm hysteresis (%), DVL: Deviation limit value (%)

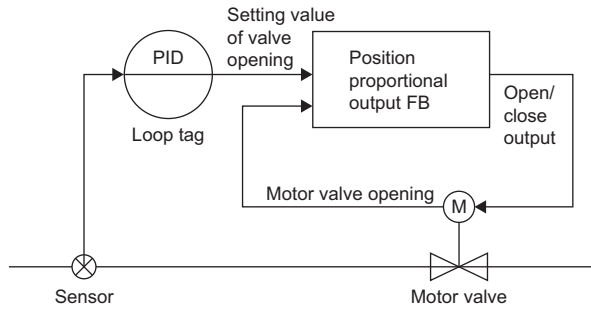
### ■PV tracking function

To avoid sudden changes of the manipulated value at mode switching (MAN → AUT/CAS/CSV), this function block matches the setting value (target) with the process variable when the control mode is MAN and keeps the value.

Condition	PV tracking processing
When PVTRK_EN is TRUE and the control mode is "MAN"	Setting value (target) = Process variable
When PVTRK_EN is FALSE and the control mode is other than "MAN"	Non-processing

### ■Proportional output operation

To perform control with a motor valve, use this function block with a loop tag FB as shown below.



This function block outputs open/close command bits to operate the motor valve opening according to the deviation (of opening) between the motor valve opening (PV) and the setting value of valve opening (current) (SVC). This function block turns on or off command signals according to the following ON/OFF conditions of command signals.

- ON/OFF condition of command signal output

The output ON/OFF conditions are calculated as shown in the following figures.

Output	Condition	Result	Figure
Output of Open Command Signal (OPEN)	$SVC - PV > DBND$	Open command signal (OPEN): OFF → ON	
	$SVC - PV \leq DBND - HS0$	Open command signal (OPEN): ON → OFF	
Output of Close Command Signal (CLOSE)	$SVC - PV < DBND$	Close command signal (CLOSE): OFF → ON	
	$SVC - PV \geq -(DBND - HS0)$	Close command signal (CLOSE): ON → OFF	

Applying a dead band and hysteresis avoids switching ON/OFF of command signal output frequently.

- Dead band (DBND)

When the motor valve opening (PV) and setting value of valve opening (current) (SVC) change frequently, ON/OFF of command signal output switches frequently.

Setting the dead band (DBND) avoids frequent switching of command signal output.

DBND: Deviation of opening at which open/close command signals are output. (The command signals are not output until the deviation of opening becomes larger than DBND.)

- Hysteresis (HS0)

Using hysteresis enables the motor valve to open by the amount of the opening set with the hysteresis even though the deviation of opening is less than or equals to the dead band.

DBND-HS0: Command signal output stops when the deviation of opening falls within the range of DBND-HS0 during the command signal output.

## ■Output at high or low limit of valve opening

The following operations can be selected with the output enabled at high/low limit for opening (LMTOUT\_EN) for when PV (motor valve opening) is 0% or less, or 100% or more during command signal output: output the result of a proportional output operation as it is or turn off the output forcibly.

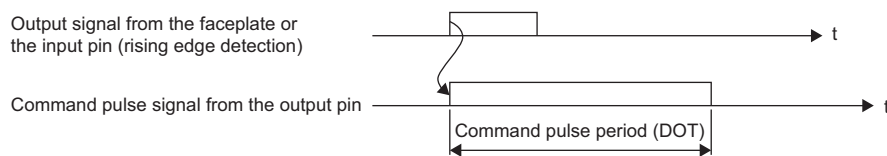
TRUE: When PV is 0 or 100%, the result of a proportional output operation is output as it is.

FALSE: When PV is 0%, CLOSE is set to FALSE. When PV is 100%, OPEN is set to FALSE.

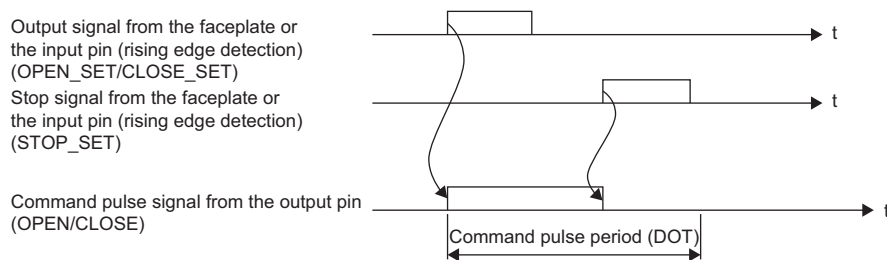
## ■One-shot output for command pulse period

This function block performs one-shot output for command pulse period according to the operation from the faceplate or the input from the input variables (OPEN\_SET, CLOSE\_SET).

- When the setting is performed with the faceplate or the value of the input variable (OPEN\_SET) is changed from FALSE to TRUE, this function block outputs a command pulse signal (TRUE) from the output variable OPEN by the time period set with the command pulse period (DOT).
- When the setting is performed with the faceplate or the value of the input variable (CLOSE\_SET) is changed from FALSE to TRUE, this function block outputs a command pulse signal (TRUE) from the output variable CLOSE by the time period set with the command pulse period (DOT).



- When stop operation is performed with the faceplate or the value of the input variable (STOP\_SET) is changed from FALSE to TRUE, this function block turns off the command pulse signal being output from the output variable OPEN or CLOSE.

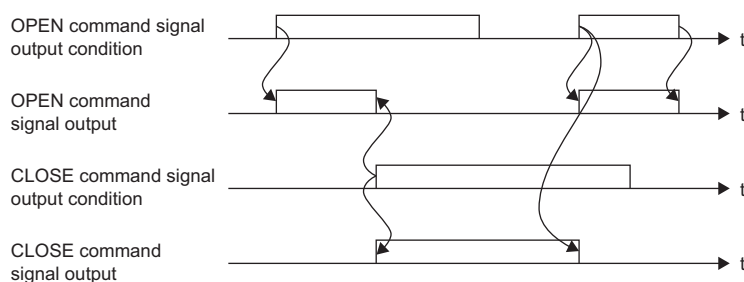


## ■Output exclusion processing

When both the OPEN command output condition and the CLOSE command output condition are satisfied, only the command output whose output condition is satisfied later is set to TRUE. (The other output command is set to FALSE.)

### Ex.

When the command signal output condition of output variable OPEN or CLOSE is TRUE simultaneously



## ■Operation location input

When TRUE is input to the LOCAL pin, a proportional output operation is not performed and the outputs from the output pins OPEN and CLOSE of the FB are set to FALSE. When the input to the LOCAL pin is changed from FALSE to TRUE, the control mode switches to the MANUAL mode.

## ■ Tracking processing

The following table shows whether tracking processing to the input variable CASIN\_T is performed or not.

Condition		Result
Tracking Flag (TRK)	Setting Value (SV) Used (SVPTN_B0)	
1	FALSE	The input variable CASIN_T is tracked.
	TRUE	The input variable CASIN_T is not tracked.
0	FALSE or TRUE	

## ■ SV variation rate & high/low limiter

This function block checks the variation rate and high/low limits to the setting value of valve opening (target) (SV) every execution cycle.

- Variation rate limiter

The SV variation rate high limit value input in percentage is processed.

Condition	Variation rate limiter result	Alarm 2 (ALM2) SV Variation Rate Limit Alarm (DSVLA)
$ SV - SVC  \leq DSVL$	SV	FALSE (Reset)
$SV - SVC > DSVL$	$SVC + DSVL$	TRUE (Detected)
$SV - SVC < DSVL$	$SVC - DSVL$	TRUE (Detected)

SV: Setting value of valve opening (target), SVC: Setting value of valve opening (current)

- High/low limiter

When SVLMT\_EN is TRUE

Condition	High/low limiter result	Alarm 2 (ALM2) SV Variation Rate Limit Alarm (DSVLA)	
		SV low limit (SVLA)	SV high limit (SVHA)
Variation rate limiter result > SH	SH	FALSE (Reset)	TRUE (Detected)
Variation rate limiter result < SL	SL	TRUE (Detected)	FALSE (Reset)
$SL \leq \text{Variation rate limiter result} \leq SH$	Variation rate limiter result	FALSE (Reset)	FALSE (Reset)

The high/low limiter result is stored in the setting value of valve opening (current) (SVC).

When SVLMT\_EN is FALSE

The variation rate limiter result is stored in the setting value of valve opening (current) (SVC).

## ■ Time-out detection/time-out reset

- Time-out detection

A time-out (TOA) of an alarm (ALM) will occur if TRUE is not input from the status answer input (OPEN\_ST/CLOSE\_ST) for a time longer than the time set with the time-out timer (TOT) after a command signal (TRUE) is output from the output variable OPEN or CLOSE.

Condition	Alarm
	Time-out (TOA)
Time to a status answer signal input $\geq$ Time set with the time-out timer (TOT)	TRUE (Detected)
Time to a status answer signal input < Time set with the time-out timer (TOT)	FALSE (Reset)

- Time-out reset

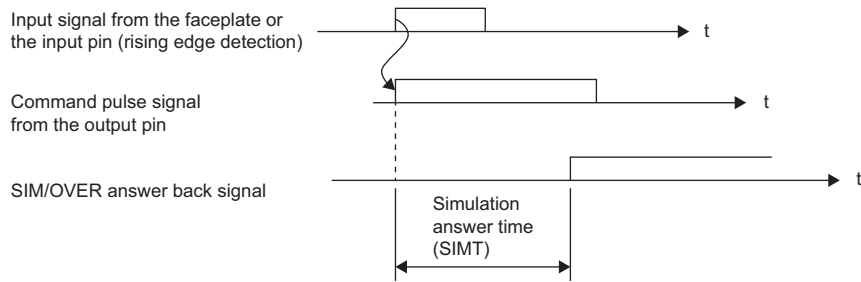
A time-out (TOA) of an alarm (ALM) is reset (FALSE) by either of the following operations.

- A command pulse signal is output from the output variable (OPEN, CLOSE) by the operation with the faceplate or the input from the input variable (OPEN\_SET, CLOSE\_SET).
- TRUE is input to the input variable (TO\_RST).

## ■SIM/OVER answer back time delay signal

In SIMULATION mode or OVERRIDE mode, this function block creates a status answer signal after a command signal output in the CPU module.

The lag time of this status answer signal is set with the simulation answer time (SIMT).



## ■Disable alarm detection

Set whether to detect the alarm (ALM) and alarm 2 (ALM2) or not in the SV variation rate & high/low limiter and time-out detection.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	<p>If the following bits in the disable alarm detection (INH) and disable alarm detection 2 (INH2) are TRUE, TOA, TRIPA, DSVLA, SVHA, and SVLA of the alarm (ALM) and the alarm 2 (ALM2) will not be detected.</p> <ul style="list-style-type: none"> <li>• TOI</li> <li>• TRIPI</li> <li>• ERRI</li> <li>• DSVLI</li> <li>• SVHI</li> <li>• SVLI</li> </ul>
Disable alarm detection by loop stop processing	<p>Page 508 Loop stop processing</p>

## ■Output processing at occurrence of SEA

The function block sets both OPEN and CLOSE of command signal output to FALSE when a sensor error (SEA) has occurred in M+P\_IN, a tag access FB.

## ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) or the tag stop (TSTP) of the monitor output buffer (DOM) is TRUE, this function block performs the following processing.

- Both OPEN and CLOSE output FALSE.
- The control mode is automatically changed to MANUAL.
- The trip (TRIPA) and time-out (TOA) are not reset.
- When DSVLA, SVHA, or SVLA of the alarm 2 (ALM2) has occurred, the DSVLA, SVHA, or SVLA is reset.
- No alarm is detected in the SV variation rate & high/low limiter.

## ■ Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation				
	Deviation check	PV tracking	Proportional output operation	Output at high or low limit of valve opening	One-shot output for command pulse period
MAN	○	○	×	×	○
AUT	○	×	○	○	×
CAS, CSV	○	×	○	○	×

Control mode	Processing operation				
	Output exclusion processing	Tracking	SV variation rate & high/low limiter	Time-out detection/ time-out reset	Alarm
MAN	○	○*1	○*2	○	○*3
AUT	○	○*1	○	○	○*4
CAS, CSV	○	×	○	○	○*4

\*1 Tracking is performed when the tracking flag (TRK) is 1.

\*2 In the MAN mode, SV variation rate limiter processing is not performed.

\*3 An alarm (ALM) whose corresponding bit is TRUE (detected) is reset, and the alarm will not be detected.

\*4 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

## Operation error

Error code (SD0)	Description
3402H	Input data (DVL, DVLS, SVC, PV, SV) or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

An error may occur in a function block that is used in a tag FB. For details on the error, refer to corresponding function blocks.

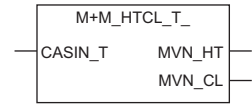
# 17.40 Heating and Cooling Output (M+M\_HTCL\_T\_)

## M+M\_HTCL\_T\_

This FB outputs two manipulated values after split conversion and output conversion to setting values. It can perform temperature control by outputting the values to both the heating and cooling operation terminal.

The primary loop can be tracked in the cascade connection. If output open, stop alarm, tag stop, or control mode change occurs, the primary loop control mode is switched to MANUAL.

### FBD/LD



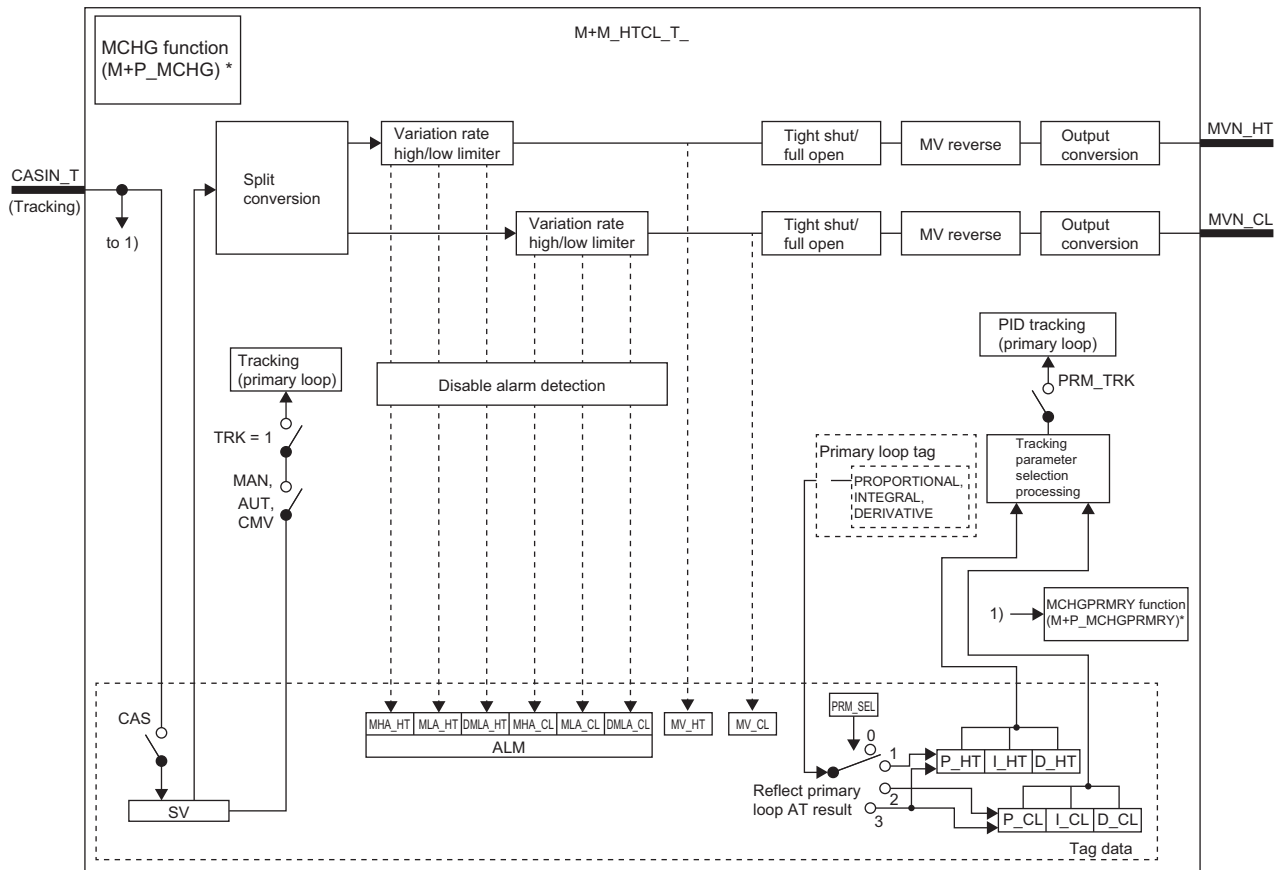
### Applicable tag type

HTCL

### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Block diagram



\* (M+P\_□) indicates the main structure member tag access FB which is the main component of each function.



■Input/output variable

Variable name	Description	Recommended range	Type	Data type
CASIN_T	Cascade SV input (unit: %) (With tracking) (Indirect address)	0 to 100[%]	Input variable	DWORD
MVN_HT	Output to a module (heating side)	NMIN_HT to NMAX_HT	Output variable	REAL
MVN_CL	Output to a module (cooling side)	NMIN_CL to NMAX_CL	Output variable	REAL

■Public variable (operation constant)

Variable name	Description	Recommended range	Initial value	Set by	Data type
NMAX_HT	Heating Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
NMIN_HT	Heating Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
MVREV_HT_EN	Heating MV Reverse Execution condition	TRUE: Valid FALSE: Invalid	FALSE	User	BOOL
NMAX_CL	Cooling Output Conversion High Limit	-999999 to 999999	100.0	User	REAL
NMIN_CL	Cooling Output Conversion Low Limit	-999999 to 999999	0.0	User	REAL
MVREV_CL_EN	Cooling MV Reverse Execution condition	TRUE: Valid FALSE: Invalid	FALSE	User	BOOL
FOTS_HT_EN	Heating Tight Shut/Full Open Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
MVFO_HT	Heating Output Value for Full Open (unit: %)	100 to 125[%]	112.5	User	REAL
MVTS_HT	Heating Output Value for Tight Shut (unit: %)	-25 to 0[%]	-16.82	User	REAL
FOTS_CL_EN	Cooling Tight Shut/Full Open Execution condition	TRUE: Execute FALSE: Stop	FALSE	User	BOOL
MVFO_CL	Cooling Output Value for Full Open (unit: %)	100 to 125[%]	112.5	User	REAL
MVTS_CL	Cooling Output Value for Tight Shut (unit: %)	-25 to 0[%]	-16.82	User	REAL
TRK	Tracking Flag <sup>*1</sup>	0: Not executed 1: Executed	0	User	INT
SVPTN_B0	Setting Value (SV) Used	TRUE: Not used FALSE: Used	TRUE	User	BOOL
SVPTN_B1	Setting Value (SV) Pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	User	BOOL
HBOTIME	Heater Burnout Detecting Time (Sec)	0 to 99999999[s] 0: Invalid 1 to 99999999: Detection time	0	User	DINT
TEMPALM_EN	Temperature Anomaly Detection Execution condition	TRUE: Valid FALSE: Invalid	FALSE	User	BOOL
MCHG_BNAL_MODE <sup>*2*3</sup>	Processing selection when output open occurs	0: Comply with the setting common to all tags 1: Switch to MANUAL mode 2: Do not switch to MANUAL mode	0	User	INT
MCHGPRMRY_OOA_EN <sup>*2</sup>	When output open occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_SPA_EN <sup>*2</sup>	When a stop alarm occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_TSTP_EN <sup>*2</sup>	When tag stop occurs, switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL
MCHGPRMRY_CASCASDR_EN <sup>*2</sup>	When the control mode is changed (CAS→AUT/MAN/CSV/CMV), switches the primary loop to MANUAL mode <sup>*4</sup>	TRUE: Switch FALSE: Do not switch	FALSE	User	BOOL

\*1 When 1 (tracking is executed) is set to the tracking flag, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

\*2 This public variable was added in version 1.070Y of the function block.

\*3 If the version of the function block is lower than 1.070Y, complies with the setting common to all tags.

\*4 When TRUE (switch) is set, connect CASOUT\_T of the primary loop with the input variable CASIN\_T.

## ■Public variable (others) \*1

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1 to 5 1: MAN 2: AUT 3: CAS 4: CMV 5: CSV	1	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

## ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

☞ Page 879 HTCL

## ■Function of the components

This tag FB consists of the following function blocks.

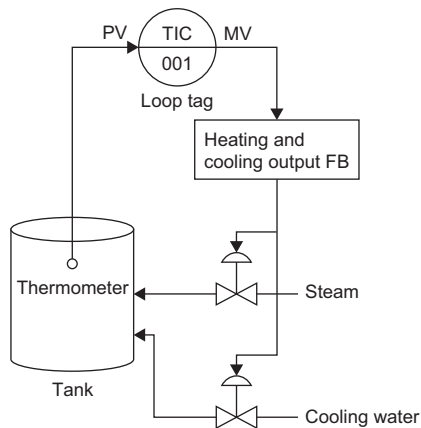
Item	Function block	Reference
MCHG function	M+P_MCHG	Page 378 M+P_MCHG
MCHGPRMRY function*1	M+P_MCHGPRMRY	Page 381 M+P_MCHGPRMRY

\*1 This function was added in version 1.070Y of the function block.

## Processing details

### ■Split conversion

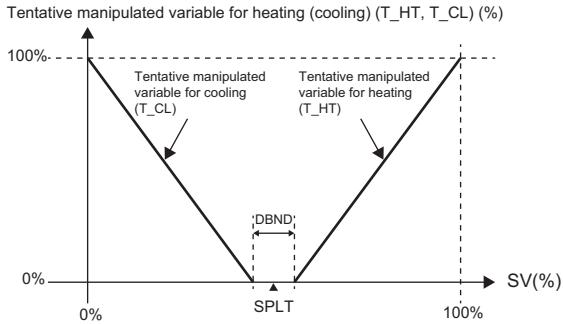
The function block is used to control two (heating and cooling) operation terminals with a loop tag FB as shown in the following figure.



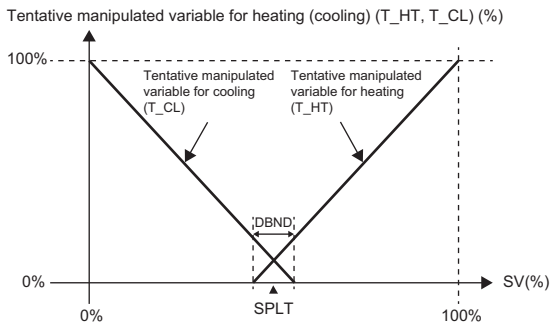
• Split conversion

This function block performs split conversion for the setting value (SV) and calculates each tentative manipulated value for heating and cooling (T<sub>HT</sub>, T<sub>CL</sub>).

Dead band (DBND) ≥ 0



Dead band (DBND) < 0



Split median (SPLT)

A value at which output is switched between heating output and cooling output in split conversion.

Dead band (DBND)

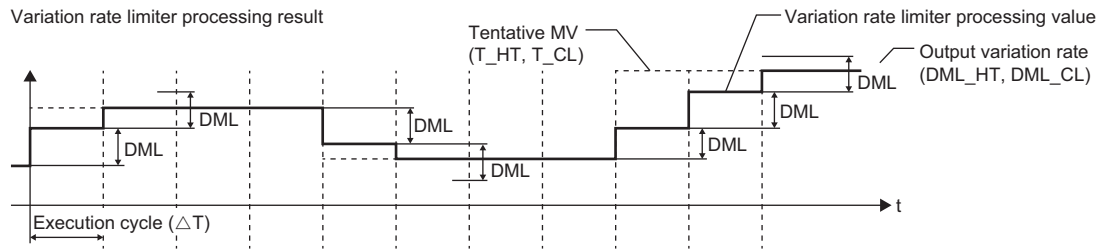
When the setting value (SV) is within a half range from the split median (SPLT) to the value set in the dead band (DBND), the tentative manipulated value for heating or cooling (T<sub>HT</sub>, T<sub>CL</sub>) will be the low limit value.

When a negative value is set in the dead band (DBND) and the setting value (SV) exceeds a half range from the split median (SPLT) to the value set in the dead band (DBND), the tentative manipulated value for heating or cooling (T<sub>HT</sub>, T<sub>CL</sub>) will be the low limit value.

## Variation rate & high/low limiter

This function block checks the variation rate and the high/low limit of each tentative manipulated value for heating and cooling ( $T_{HT}$ ,  $T_{CL}$ ) after split conversion.

- Variation rate limiter



### Variation rate limiter for heating

Condition	Variation rate limiter processing result	Alarm (ALM)
		Output variation rate limit (DMLA_HT)
$ T_{HT} - MV_{HT}  \leq DML_{HT}$	$T_{HT}$	FALSE (Reset)
$T_{HT} - MV_{HT} > DML_{HT}$	$MV_{HT} + DML_{HT}$	TRUE (Detected)
$T_{HT} - MV_{HT} < DML_{HT}$	$MV_{HT} - DML_{HT}$	TRUE (Detected)

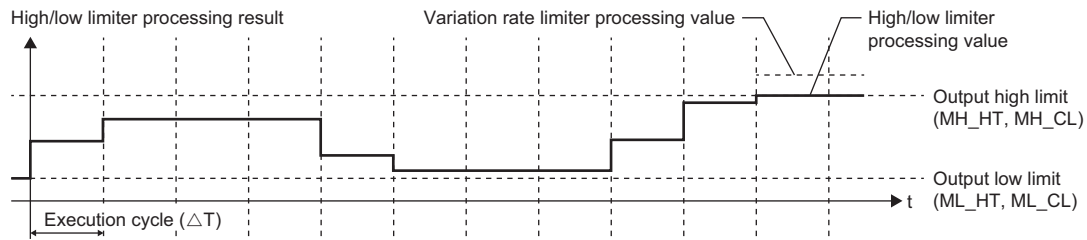
$T_{HT}$ : Tentative manipulated value for heating,  $MV_{HT}$ : Heating manipulated variable,  $DML_{HT}$ : Heating output variation rate high limit value

### Variation rate limiter for cooling

Condition	Variation rate limiter processing result	Alarm (ALM)
		Output variation rate limit (DMLA_CL)
$ T_{CL} - MV_{CL}  \leq DML_{CL}$	$T_{CL}$	FALSE (Reset)
$T_{CL} - MV_{CL} > DML_{CL}$	$MV_{CL} + DML_{CL}$	TRUE (Detected)
$T_{CL} - MV_{CL} < DML_{CL}$	$MV_{CL} - DML_{CL}$	TRUE (Detected)

$T_{CL}$ : Tentative manipulated value for cooling,  $MV_{CL}$ : Cooling manipulated variable,  $DML_{CL}$ : Cooling output variation rate high limit value

- High/low limiter



### Variation rate limiter for heating

Condition	High/low limiter processing result	Alarm (ALM)	
		Output low limit alarm (MLA_HT)	Output high limit alarm (MHA_HT)
Result of variation rate limiter processing for heating $> MH_{HT}$	$MH_{HT}$	FALSE (Reset)	TRUE (Detected)
Result of variation rate limiter processing for heating $< ML_{HT}$	$ML_{HT}$	TRUE (Detected)	FALSE (Reset)
$ML_{HT} \leq$ Result of variation rate limiter processing for heating $\leq MH_{HT}$	Variation rate limiter processing value for heating	FALSE (Reset)	FALSE (Reset)

$MH_{HT}$ : Output high limit value for heating,  $ML_{HT}$ : Output low limit value for heating

### Variation rate limiter for cooling

Condition	High/low limiter processing result	Alarm (ALM)	
		Output low limit alarm (MLA_CL)	Output high limit alarm (MHA_CL)
Result of variation rate limiter processing for cooling $> MH_{CL}$	$MH_{CL}$	FALSE (Reset)	TRUE (Detected)
Result of variation rate limiter processing for cooling $< ML_{CL}$	$ML_{CL}$	TRUE (Detected)	FALSE (Reset)
$ML_{CL} \leq$ Result of variation rate limiter processing for cooling $\leq MH_{CL}$	Variation rate limiter processing value for cooling	FALSE (Reset)	FALSE (Reset)

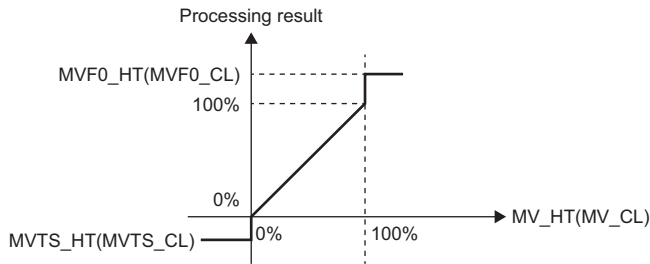
$MH_{CL}$ : Output high limit for cooling,  $ML_{CL}$ : Output low limit for cooling

### ■Tight shut/full open

This tight shut/full open function is used to completely open or close a control valve.

The processing result is reduced to the output value for tight shut when MV\_HT or MV\_CL is 0% or less, and the result is increased to the output value for full open when a MV\_HT or MV\_CL is 100% or more.

- When FOTS\_HT\_EN or FOTS\_CL\_EN is TRUE



MVTS\_HT: Heating output value for tight shut (%)  
 MVTS\_CL: Cooling output value for tight shut (%)  
 MVFO\_HT: Heating output value for full open (%)  
 MVFO\_CL: Cooling output value for full open (%)

### ■MV reverse

This function block performs inversion processing (100 - MV) to a manipulated value.

- MV reverse for heating

Condition	Processing result
MVREV_HT_EN = TRUE	MVREV_HT = 100 - MV_HT
MVREV_HT_EN = FALSE	MVREV_HT = MV_HT

MVREV\_HT: Heating output after processing of MV reverse for internal operation (%), MV\_HT: Heating manipulated variable (%)

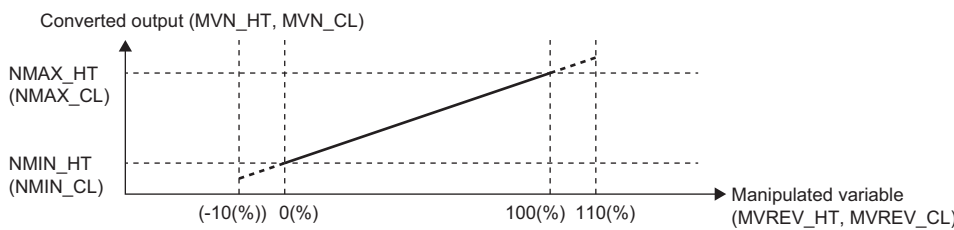
- MV reverse for cooling

Condition	Processing result
MVREV_CL_EN = TRUE	MVREV_CL = 100 - MV_CL
MVREV_CL_EN = FALSE	MVREV_CL = MV_CL

MVREV\_CL: Cooling output after processing of MV reverse for internal operation (%), MV\_CL: Cooling manipulated variable (%)

### ■Output conversion

This function block performs output conversion processing.



$$\text{Converted output (MVN\_HT)} = \{(NMAX\_HT - NMIN\_HT) \times \frac{MVREV\_HT}{100}\} + NMIN\_HT$$

$$\text{Converted output (MVN\_CL)} = \{(NMAX\_CL - NMIN\_CL) \times \frac{MVREV\_CL}{100}\} + NMIN\_CL$$

NMAX\_HT: Heating output conversion high limit value  
 NMIN\_HT: Heating output conversion low limit value  
 MVREV\_HT: Heating output after processing of MV reverse for internal operation (%)  
 MVN\_HT: Heating output conversion output value,  
 NMAX\_CL: Cooling output conversion high limit value  
 NMIN\_CL: Cooling output conversion low limit value  
 MVREV\_CL: Cooling output after processing of MV reverse for internal operation (%)  
 MVN\_CL: Cooling output conversion output value

## ■Tracking processing

The following table shows whether tracking processing to the input variable CASIN\_T is performed or not.

Condition		Result
Tracking Flag (TRK)	Setting Value (SV) Used (SVPTN_B0)	
1	FALSE	The input variable CASIN_T is tracked.
	TRUE	The input variable CASIN_T is not tracked.
0	FALSE or TRUE	

## ■Auto tuning result reflection function

This function block reflects the auto tuning result in the primary loop to tag data of this tag FB.


The following shows the procedure to calculate PID parameters for heating and cooling with auto tuning in the primary loop.

1. Set the control mode of this tag to the CAS mode.
2. Set the target to reflect results of auto tuning (PRM\_SEL) to Heating (1).
3. Perform auto tuning for heating in the primary loop. When auto tuning in the primary loop is completed successfully, the result of the auto tuning is reflected to the PID parameter for heating of this tag FB.
4. Set the target to reflect results of auto tuning (PRM\_SEL) to Cooling (2).
5. Perform auto tuning for cooling in the primary loop. When auto tuning in the primary loop is completed successfully, the result of the auto tuning is reflected to the PID parameter for cooling of this tag FB.

Condition		Tag data that stores the result of auto tuning
SVPTN_B0 = FALSE and SVPTN_B1 = FALSE	PRM_SEL = 0	Not stored.
	PRM_SEL = 1	PID parameters for heating (P_HT, I_HT, D_HT)
	PRM_SEL = 2	PID parameters for cooling (P_CL, I_CL, D_CL)
	PRM_SEL = 3	PID parameters for heating and cooling (P_HT, I_HT, D_HT, P_CL, I_CL, D_CL)

PRM\_SEL: Target to reflect results of auto tuning

For details on the auto tuning, refer to the following.

 Page 916 Auto tuning

### Point

When the auto tuning result reflection function is used, the tag types connectable as primary loops are as follows.

- PID
- 2PID
- 2PIDH

When PID parameters are tracked, the parameter values stored in this tag FB are not restored even though "Restore" is performed with the PX Developer monitor tool after auto tuning is completed in the primary loop.

Perform the following operations to restore the PID constants to the one before the auto tuning.

- Set the PID parameter tracking flag (PRM\_TRK) of this tag FB to "Not executed" (0).
- Set the PID parameter before the auto tuning in the primary loop for the PID parameter of this tag FB.
- Set the PID parameter tracking flag (PRM\_TRK) of this tag FB to "Executed" (1).

### Tracking (PID parameter)

When all the following conditions for tracking PID parameters are satisfied, this function block tracks PID parameters for heating and cooling to the primary loop.

- PRM\_TRK = 1
- SVPTN\_B0 = FALSE
- SVPTN\_B1 = FALSE

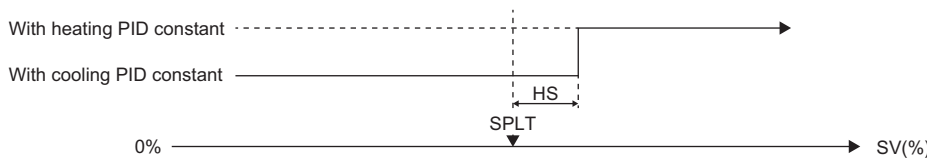
The following table shows the description on the processing.

The PID parameter to be tracked is selected with the following expressions.

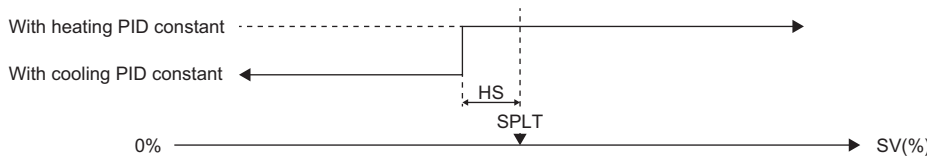
Condition	PID parameter to be tracked
PID parameter for cooling is being tracked: $SV \geq SPLT + HS$	PID parameters for heating (P_HT, I_HT, D_HT)
PID parameter for cooling is being tracked: $SV < SPLT + HS$	PID parameters for cooling (P_CL, I_CL, D_CL)
PID parameter for heating is being tracked: $SV \geq SPLT - HS$	PID parameters for heating (P_HT, I_HT, D_HT)
PID parameter for heating is being tracked: $SV < SPLT - HS$	PID parameters for cooling (P_CL, I_CL, D_CL)

SV: Primary loop MV, SPLT: Split median, HS: Hysteresis (%)

- Switching from PID parameter for cooling to PID parameter for heating



- Switching from PID parameter for heating to PID parameter for cooling



The following table lists the operations when the condition for tracking the PID parameters shown above are satisfied for the first time.

Condition	PID parameter to be tracked
$SV \geq SPLT$	PID parameters for heating (P_HT, I_HT, D_HT)
$SV < SPLT$	PID parameters for cooling (P_CL, I_CL, D_CL)

**Point**

- When the tag type of the primary loop is PID, 2PID, 2PIDH, PIDP, SPI, IPD, or BPI, the PID parameters are tracked. (When the tag type of the primary loop is SPI or BPI, the parameters to be tracked are "PROPORTIONAL" (+52) and "INTEGRAL" (+54).)
- To track PID parameters, set initial values in the PID parameters for heating and cooling of this tag FB. Do not set the PID parameters in the primary loop.
- Switching timing of PID parameters can be adjusted with a hysteresis.

## ■Disable alarm detection

Set whether to detect alarms (ALM) or not in the variation rate & high/low limiter and heater burnout detection.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, DMLA_HT, MHA_HT, MLA_HT, DMLA_CL, MHA_CL, MLA_CL, or HBOA of the alarm (ALM) will not be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• DMLI_HT</li> <li>• MHI_HT</li> <li>• MLI_HT</li> <li>• DMLI_CL</li> <li>• MHI_CL</li> <li>• MLI_CL</li> <li>• HBOI</li> </ul>
Disable alarm detection by control mode selection	In MAN or CMV mode, DMLA_HT, MHA_HT, MLA_HT, DMLA_CL, MHA_CL, MLA_CL, and HBOA of the alarm (ALM) are reset and the DMLA_HT, MHA_HT, MLA_HT, DMLA_CL, MHA_CL, MLA_CL, or HBOA will not be detected.
Disable alarm detection by loop stop processing	☞ Page 518 Loop stop processing

## ■Heater burnout detection

HBOA occurs if the manipulated value for heating is at the high limit value (MH\_HT) for a time period longer than the time period specified with HBOTIME.

HBOA is reset when the manipulated value for heating falls below the high limit value (MH\_HT).

## ■Temperature error detection in the primary loop

When TEMPALM\_EN is TRUE and HHA (input high high limit alarm) has occurred in the primary loop, this function block sets the manipulated value for heating to the low limit value of the manipulated value for heating (ML\_HT).

## ■Loop stop processing

When the stop alarm (SPA) of the alarm (ALM) or the tag stop (TSTP) of the monitor output buffer (DOM) is TRUE, this function block performs the following processing.

- Outputs (MVN\_HT, MVN\_CL) are held.
- The control mode is automatically changed to MANUAL.
- When DMLA\_HT, MHA\_HT, MLA\_HT, DMLA\_CL, MHA\_CL, MLA\_CL, or HBOA of the alarm (ALM) has occurred, the DMLA\_HT, MHA\_HT, MLA\_HT, DMLA\_CL, MHA\_CL, MLA\_CL, or HBOA is reset.
- No alarm is detected in the variation rate & high/low limiter and heater burnout detection.

## ■Processing operation

○: Performed, ×: Not performed

Control mode	Processing operation					
	Split conversion	Variation rate & high/low limiter	Tight shut/full open	MV reverse	Tracking	Output conversion
MAN, CMV	×	×	○	○	○ <sup>*1</sup>	○
AUT	○	○	○	○	○ <sup>*1</sup>	○
CAS, CSV	○	○	○	○	×	○

Control mode	Processing operation				
	Auto tuning result reflection	Tracking of PID parameters	Alarm	Heater burnout notification	Temperature error detection in the primary loop
MAN, CMV	×	○	×	×	×
AUT	×	○	○ <sup>*3</sup>	○	×
CAS, CSV	○ <sup>*4</sup>	○	○ <sup>*3</sup>	○	○

\*1 Tracking is performed when the tracking flag (TRK) is 1.

\*2 An alarm (ALM) whose corresponding bit is TRUE (detected) is reset, and the alarm will not be detected.

\*3 An alarm whose corresponding bit is TRUE (Valid) in the disable alarm detection (INH) is not detected.

\*4 This function is performed only in the CAS mode.



## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	Input data (NMAX_HT, NMIN_HT, MV_HT, NMAX_CL, NMIN_CL, MV_CL) or tag data is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# 18 STATUS TAG

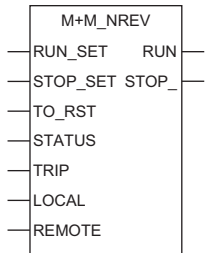
The following FBs perform reversible/irreversible operation and ON/OFF operation and operates as a timer and a counter.

## 18.1 Motor Irreversible (2 Input/2 Output) (M+M\_NREV)

### M+M\_NREV

This FB performs irreversible operation and controls a solenoid valve.

#### FBD/LD



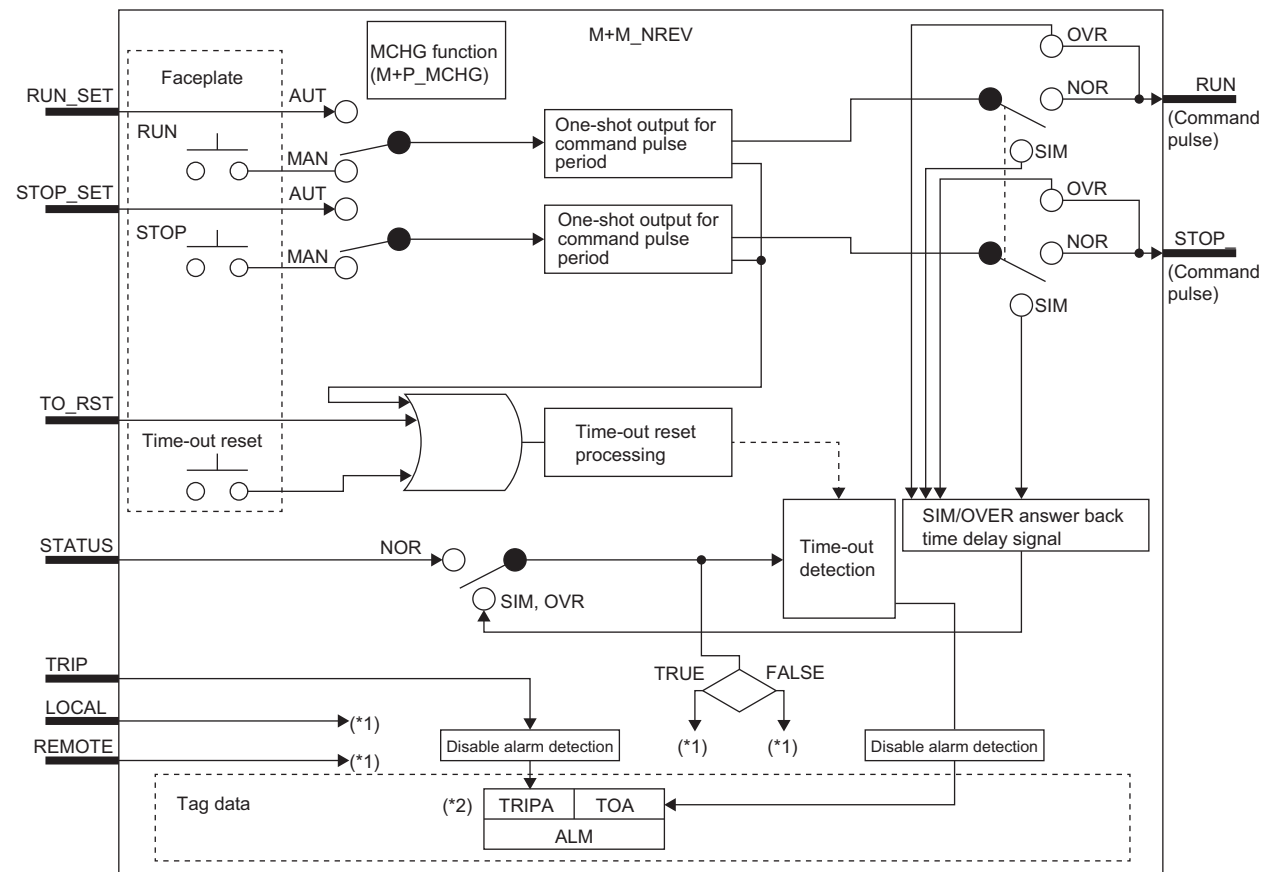
#### Applicable tag type

NREV

#### Control mode

MAN	AUT	CAS	CMV	CSV
<input type="radio"/>	<input type="radio"/>	—	—	—

#### Block diagram



\*1 For displaying the faceplate status.

\*2 Indicates bit item.

## Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
RUN_SET	External input for RUN operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: RUN	Input variable	BOOL
STOP_SET	External input for STOP operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: STOP	Input variable	BOOL
TO_RST	Time-out error external reset input	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: Time-out reset	Input variable	BOOL
STATUS	Status answer input	TRUE: RUN FALSE: STOP* <sup>1</sup>	Input variable	BOOL
TRIP	External failure (TRIP) input	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL
LOCAL	Local operation selection signal	TRUE: Valid FALSE: Invalid* <sup>1</sup>	Input variable	BOOL
REMOTE	Remote operation selection signal	TRUE: Valid FALSE: Invalid* <sup>1</sup>	Input variable	BOOL
RUN	ON output for command pulse period	TRUE: Run FALSE: —* <sup>1</sup>	Output variable	BOOL
STOP_	ON output for command pulse period	TRUE: Run FALSE: —* <sup>1</sup>	Output variable	BOOL

\*1 The default value is FALSE.

### ■Public variable (others) \*<sup>1</sup>

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1, 2 1: MAN 2: AUT	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

### ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

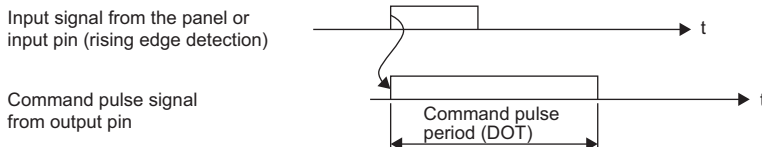
☞ Page 882 NREV

## Processing details

### ■ One-shot output for command pulse period

This function block performs one-shot output for command pulse period according to the operation from the faceplate or the input from the input variables (RUN\_SET, STOP\_SET).

- When the setting is performed with the faceplate or the value of the input variable (RUN\_SET) is changed from FALSE to TRUE, this function block outputs a command pulse signal (TRUE) from the output variable RUN by the time period set with the command pulse period (DOT).
- When the setting is performed with the faceplate or the value of the input variable (STOP\_SET) is changed from FALSE to TRUE, this function block outputs a command pulse signal (TRUE) from the output variable STOP\_ by the time period set with the command pulse period (DOT).



### ■ Time-out detection/time-out reset

- Time-out detection

A time-out (TOA) of an alarm (ALM) will occur if TRUE or FALSE is not input from the status answer input (STATUS) for a time longer than the time set with the time-out timer (TOT) after a command pulse signal (TRUE) is output from the output variable RUN or STOP\_.

Condition	Alarm
	Time-out (TOA)
Time to a status answer signal input $\geq$ Time set with the time-out timer (TOT)	TRUE (Detected)
Time to a status answer signal input $<$ Time set with the time-out timer (TOT)	FALSE (Reset)

- Time-out reset

A time-out (TOA) of an alarm (ALM) is reset (FALSE) by either of the following operations.

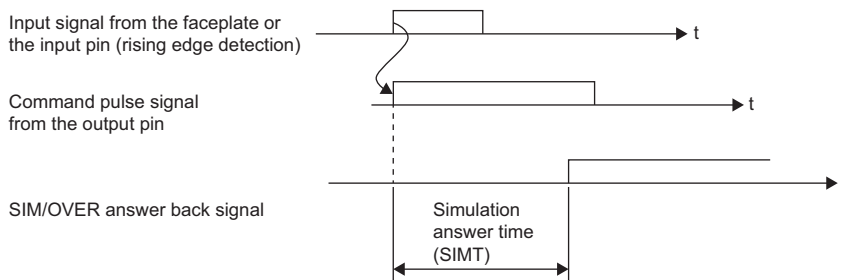
A command pulse signal is output from the output variable (RUN, STOP\_) by the operation with the faceplate or the input from the input variable (RUN\_SET, STOP\_SET).

TRUE is input to the input variable (TO\_RST).

### ■ SIM/OVER answer back time delay signal

In SIMULATION mode or OVERRIDE mode, this function block creates a status answer signal after a command signal output in the CPU module.

The lag time of this status answer signal is set with the simulation answer time (SIMT).



### ■ Disable alarm detection

This function block sets whether to detect alarms (ALM) or not.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no TRIPA and TOA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• TRIP1</li> <li>• TOI</li> </ul>

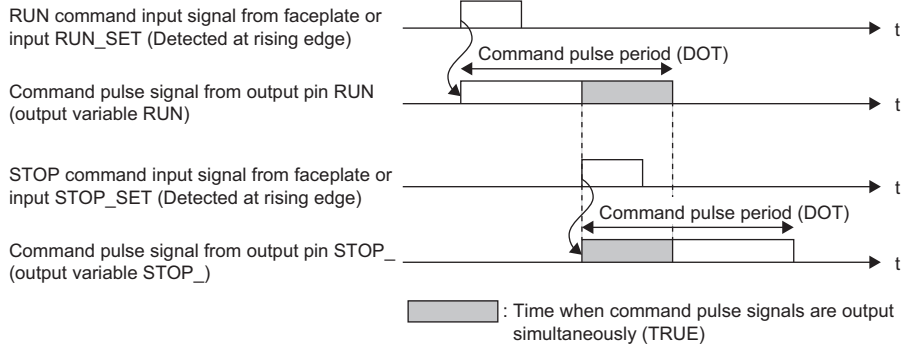
## Operation error

There is no operation error.

### Point

If the STOP command (or RUN command) is issued while a command pulse signal (TRUE) is output from the output variable RUN (or output variable STOP\_), the command pulse signals (TRUE) are output simultaneously from the output variable RUN and output variable STOP\_.

<When command pulse signals (TRUE) are output simultaneously from output variable RUN and output variable STOP\_>



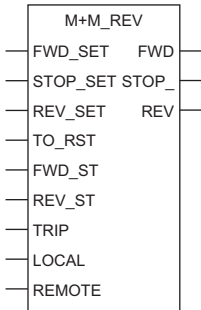
When the output variable RUN and output variable STOP\_ are output directly to the external device, multiple commands (RUN command and STOP command) may be output simultaneously to the external device. Not to output multiple commands to the external device simultaneously, create a program to output only either one of the command pulse signals to the external device.

# 18.2 Motor Reversible (2 Input/3 Output) (M+M\_REV)

## M+M\_REV

This FB performs reversible operation.

### FBD/LD



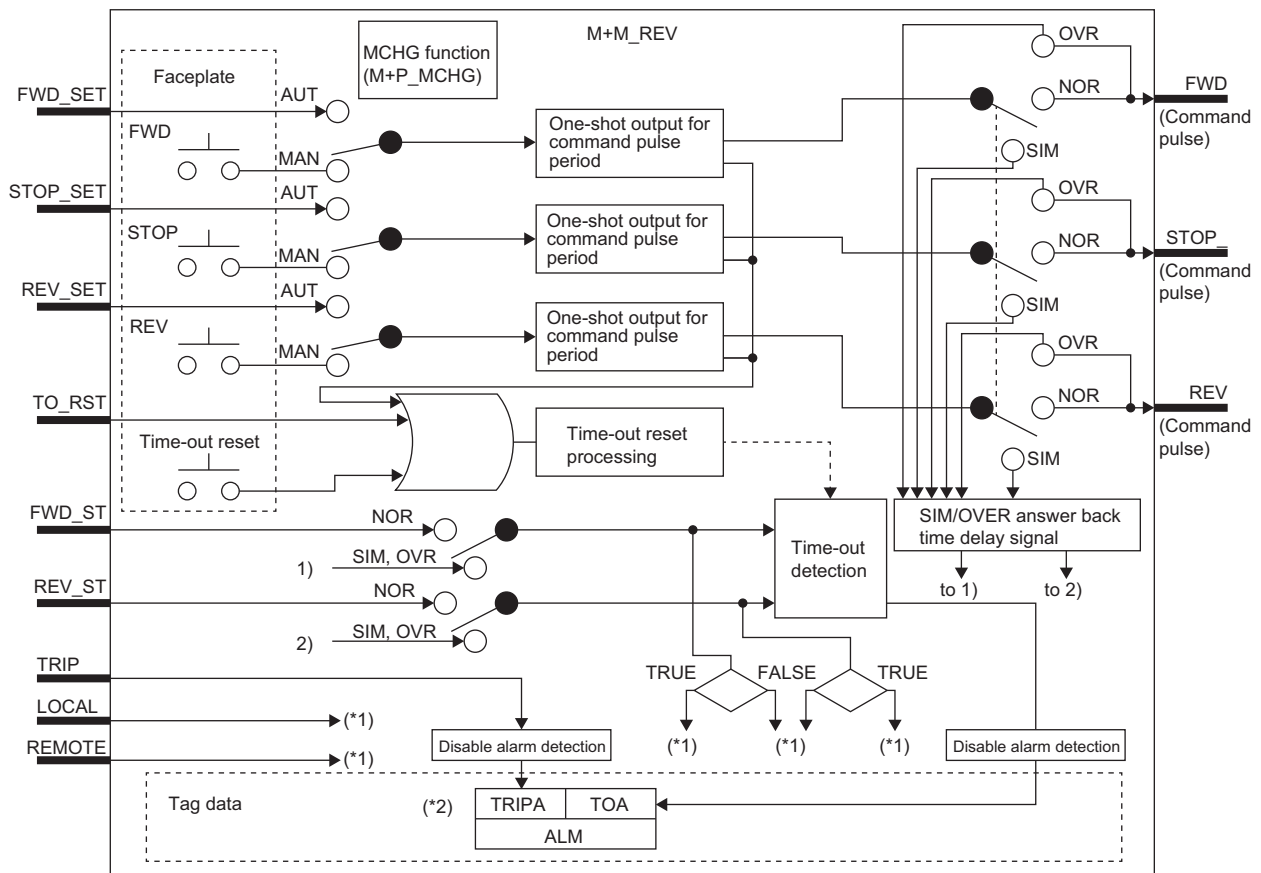
### Applicable tag type

REV

### Control mode

MAN	AUT	CAS	CMV	CSV
○	○	—	—	—

### Block diagram



\*1 For displaying the faceplate status

\*2 Indicates bit item

## Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
FWD_SET	External input of FWD (forward rotation)	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: FWD	Input variable	BOOL
STOP_SET	External input for STOP operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: STOP	Input variable	BOOL
REV_SET	External input of REV (reverse rotation)	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: REV	Input variable	BOOL
TO_RST	Time-out error external reset input	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: Time-out reset	Input variable	BOOL
FWD_ST	Status answer input	TRUE: FWD FALSE: STOP* <sup>1</sup>	Input variable	BOOL
REV_ST	Status answer input	TRUE: REV FALSE: STOP* <sup>1</sup>	Input variable	BOOL
TRIP	External failure (TRIP) input	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL
LOCAL	Local operation selection signal	TRUE: Valid FALSE: Invalid* <sup>1</sup>	Input variable	BOOL
REMOTE	Remote operation selection signal	TRUE: Valid FALSE: Invalid* <sup>1</sup>	Input variable	BOOL
FWD	ON output for command pulse period	TRUE: Run FALSE: —* <sup>1</sup>	Output variable	BOOL
STOP_	ON output for command pulse period	TRUE: Run FALSE: —* <sup>1</sup>	Output variable	BOOL
REV	ON output for command pulse period	TRUE: Run FALSE: —* <sup>1</sup>	Output variable	BOOL

\*1 The default value is FALSE.

### ■Public variable (others) \*<sup>1</sup>

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1, 2 1: MAN 2: AUT	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

### ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

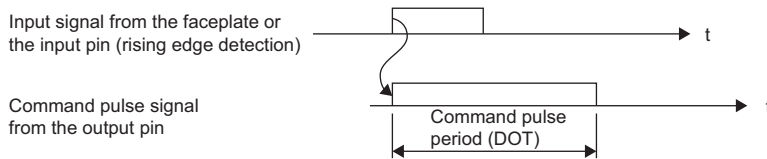
📄 Page 883 REV

## Processing details

### ■ One-shot output for command pulse period

This function block performs one-shot output for command pulse period according to the operation from the faceplate or the input from the input variables (FWD\_SET, STOP\_SET, REV\_SET).

- When the setting is performed with the faceplate or the value of the input variable (FWD\_SET) is changed from FALSE to TRUE, this function block outputs a command pulse signal (TRUE) from the output variable FWD by the time period set with the command pulse period (DOT).
- When the setting is performed with the faceplate or the value of the input variable (STOP\_SET) is changed from FALSE to TRUE, this function block outputs a command pulse signal (TRUE) from the output variable STOP\_ by the time period set with the command pulse period (DOT).
- When the setting is performed with the faceplate or the value of the input variable (REV\_SET) is changed from FALSE to TRUE, this function block outputs a command pulse signal (TRUE) from the output variable REV by the time period set with the command pulse period (DOT).



### ■ Time-out detection/time-out reset

- Time-out detection

A time-out (TOA) of an alarm (ALM) will occur if TRUE or FALSE is not input from the status answer input (FWD\_ST) for a time longer than the time set with the time-out timer (TOT) after a command pulse signal (TRUE) is output from the output variable FWD or STOP\_.

A time-out (TOA) of an alarm (ALM) will occur if TRUE or FALSE is not input from the status answer input (REV\_ST) for a time longer than the time set with the time-out timer (TOT) after a command pulse signal (TRUE) is output from the output variable REV or STOP\_.

Condition	Alarm
	Time-out (TOA)
Time to a status answer signal input $\geq$ Time set with the time-out timer (TOT)	TRUE (Detected)
Time to a status answer signal input $<$ Time set with the time-out timer (TOT)	FALSE (Reset)

- Time-out reset

A time-out (TOA) of an alarm (ALM) is reset (FALSE) by either of the following operations.

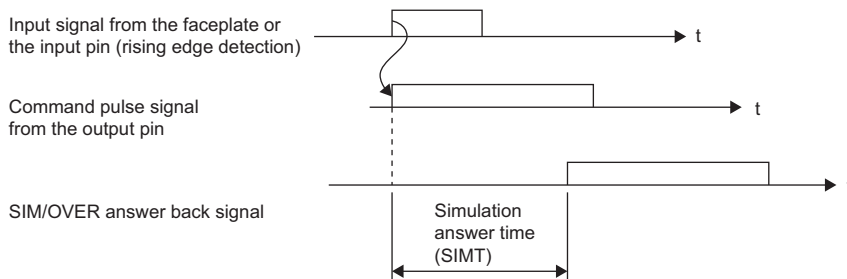
A command pulse signal is output from the output variable (FWD, STOP\_, REV) by the operation with the faceplate or the input from the input variable (FWD\_SET, STOP\_SET, REV\_SET).

TRUE is input to the input variable (TO\_RST).

### ■ SIM/OVER answer back time delay signal

In SIMULATION mode or OVERRIDE mode, this function block creates a status answer signal after a command signal output in the CPU module.

The lag time of this status answer signal is set with the simulation answer time (SIMT).





### ■Disable alarm detection

This function block sets whether to detect alarms (ALM) or not.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no TRIPA and TOA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• TRIPI</li> <li>• TOI</li> </ul>

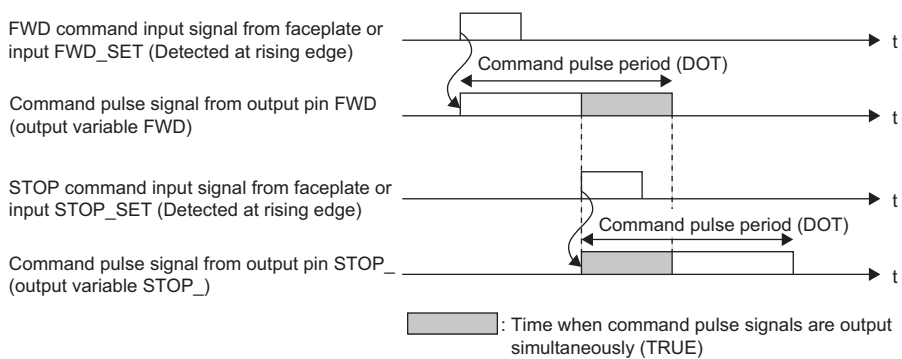
### Operation error

There is no operation error.



If another command is issued while a command pulse signal (TRUE) is output from the output variable FWD, STOP\_, or REV, the command pulse signals (TRUE) are output simultaneously.

<When multiple command pulse signals (TRUE) are output>



When the output variables FWD, STOP\_, and REV are output directly to the external device, multiple commands (FWD command, STOP command, and REV command) may be output simultaneously to the external device.

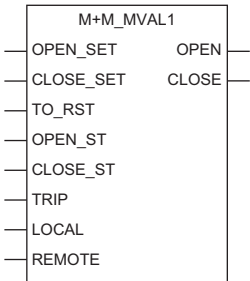
Not to output multiple commands to the external device simultaneously, create a program to output only either one of the command pulse signals to the external device.

# 18.3 ON/OFF Operation (2 Input/2 Output) (M+M\_MVAL1)

## M+M\_MVAL1

This FB controls an ON/OFF motor valve and solenoid valve.

### FBD/LD



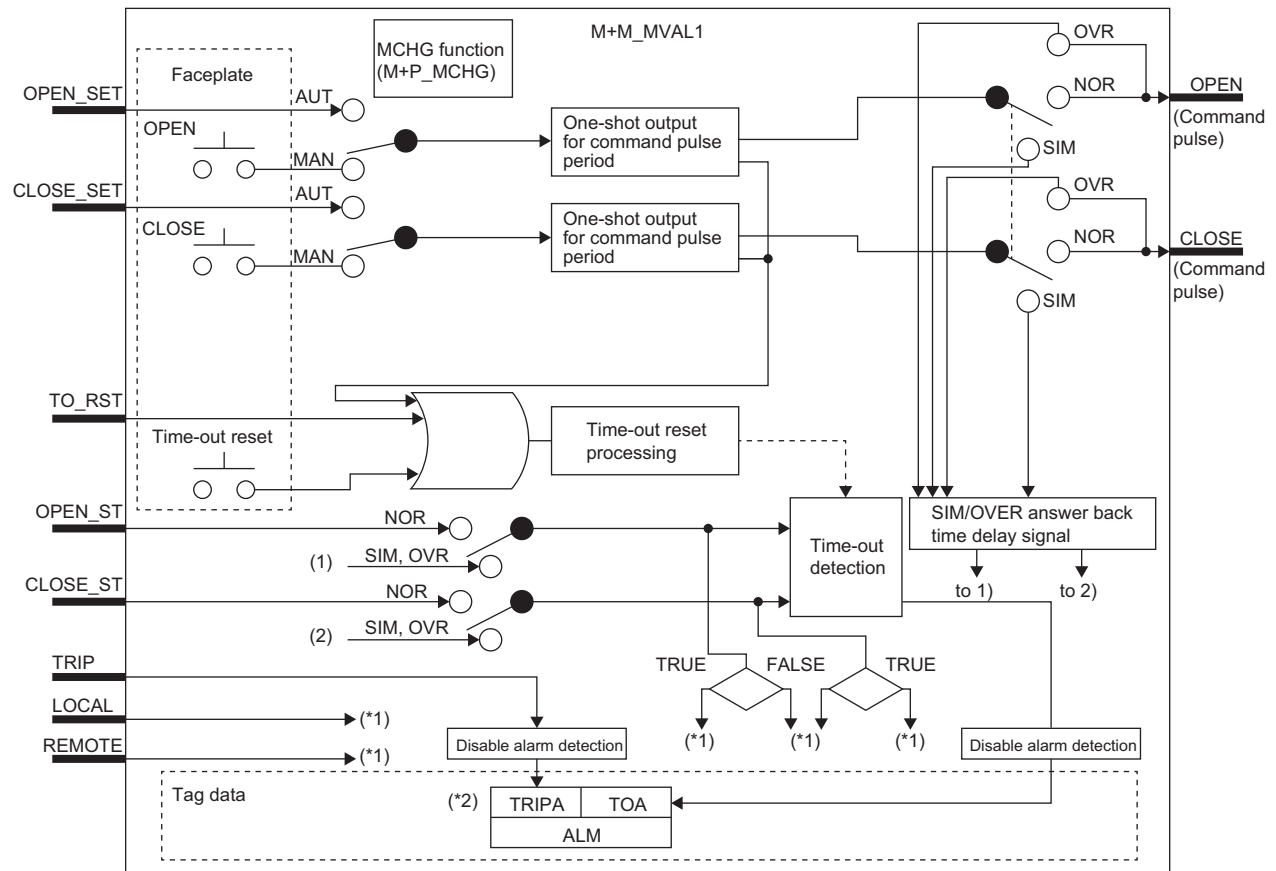
### Applicable tag type

MVAL1

### Control mode

MAN	AUT	CAS	CMV	CSV
○	○	—	—	—

### Block diagram



\*1 For displaying the faceplate status

\*2 Indicates bit item

## Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
OPEN_SET	External input for OPEN operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: OPEN	Input variable	BOOL
CLOSE_SET	External input for CLOSE operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: CLOSE	Input variable	BOOL
TO_RST	Time-out error external reset input	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: Time-out reset	Input variable	BOOL
OPEN_ST	Status answer input	TRUE: OPEN FALSE: SEMI_CLOSE* <sup>1</sup>	Input variable	BOOL
CLOSE_ST	Status answer input	TRUE: CLOSE FALSE: SEMI_CLOSE* <sup>1</sup>	Input variable	BOOL
TRIP	External failure (TRIP) input	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL
LOCAL	Local operation selection signal	TRUE: Valid FALSE: Invalid* <sup>1</sup>	Input variable	BOOL
REMOTE	Remote operation selection signal	TRUE: Valid FALSE: Invalid* <sup>1</sup>	Input variable	BOOL
OPEN	ON output for command pulse period	TRUE: Run FALSE: —* <sup>1</sup>	Output variable	BOOL
CLOSE	ON output for command pulse period	TRUE: Run FALSE: —* <sup>1</sup>	Output variable	BOOL

\*1 The default value is FALSE.

### ■Public variable (others) \*<sup>1</sup>

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1, 2 1: MAN 2: AUT	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

### ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

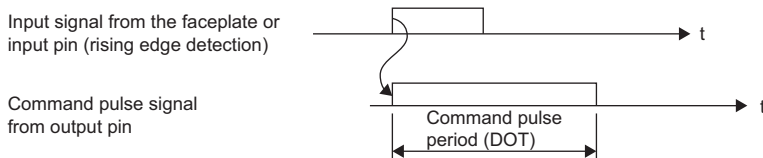
📄 Page 884 MVAL1

## Processing details

### ■ One-shot output for command pulse period

This function block performs one-shot output for command pulse period according to the operation from the faceplate or the input from the input variables (OPEN\_SET, CLOSE\_SET).

- When the setting is performed with the faceplate or the value of the input variable (OPEN\_SET) is changed from FALSE to TRUE, this function block outputs a command pulse signal (TRUE) from the output variable OPEN by the time period set with the command pulse period (DOT).
- When the setting is performed with the faceplate or the value of the input variable (CLOSE\_SET) is changed from FALSE to TRUE, this function block outputs a command pulse signal (TRUE) from the output variable CLOSE by the time period set with the command pulse period (DOT).



### ■ Time-out detection/time-out reset

- Time-out detection

A time-out (TOA) of an alarm (ALM) will occur if TRUE is not input from the status answer input (OPEN\_ST/CLOSE\_ST) for a time longer than the time set with the time-out timer (TOT) after a command pulse signal (TRUE) is output from the output variable OPEN or CLOSE.

Condition	Alarm
	Time-out (TOA)
Time to a status answer signal input $\geq$ Time set with the time-out timer (TOT)	TRUE (Detected)
Time to a status answer signal input $<$ Time set with the time-out timer (TOT)	FALSE (Reset)

- Time-out reset

A time-out (TOA) of an alarm (ALM) is reset (FALSE) by either of the following operations.

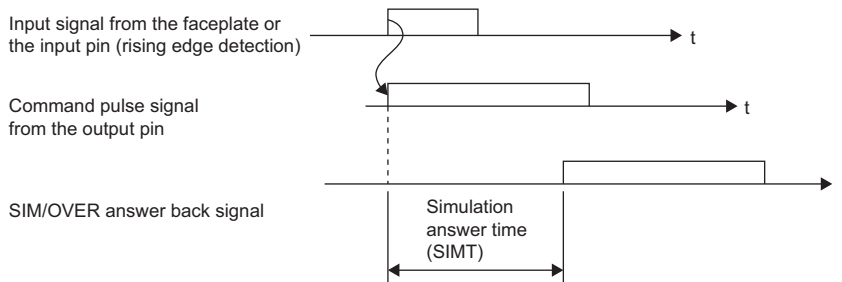
A command pulse signal is output from the output variable (OPEN, CLOSE) by the operation with the faceplate or the input from the input variable (OPEN\_SET, CLOSE\_SET).

TRUE is input to the input variable (TO\_RST).

### ■ SIM/OVER answer back time delay signal

In SIMULATION mode or OVERRIDE mode, this function block creates a status answer signal after a command signal output in the CPU module.

The lag time of this status answer signal is set with the simulation answer time (SIMT).



### ■ Disable alarm detection

This function block sets whether to detect alarms (ALM) or not.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	<p>If the following bits in the disable alarm detection (INH) are TRUE, no TRIPA and TOA of the alarm (ALM) will be detected.</p> <ul style="list-style-type: none"> <li>• ERRI</li> <li>• TRIPI</li> <li>• TOI</li> </ul>

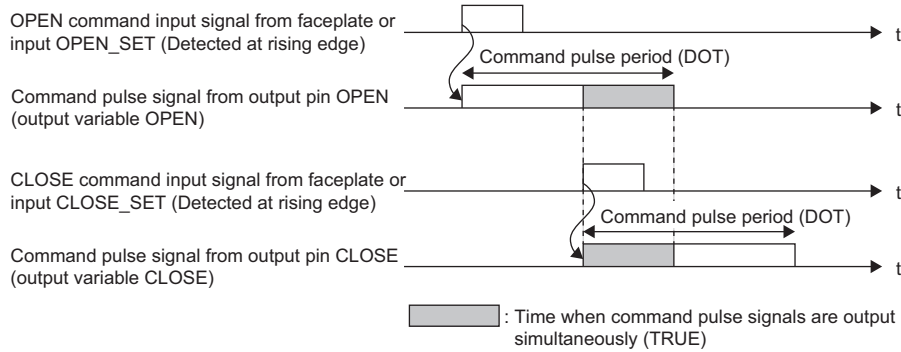
## Operation error

There is no operation error.

### Point

If the CLOSE command (or OPEN command) is issued while a command pulse signal (TRUE) is output from the output variable OPEN (or output variable CLOSE), the command pulse signals (TRUE) are output simultaneously from the output variable OPEN and output variable CLOSE.

<When command pulse signals (TRUE) are output simultaneously from output variable OPEN and output variable CLOSE>



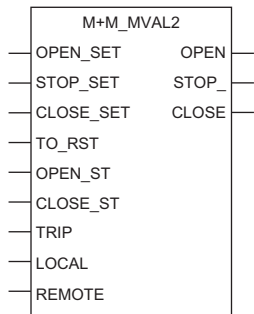
When the output variable OPEN and output variable CLOSE are output directly to the external device, multiple commands (OPEN command and CLOSE command) may be output simultaneously to the external device. Not to output multiple commands to the external device simultaneously, create a program to output only either one of the command pulse signals to the external device.

# 18.4 ON/OFF Operation (2 Input/3 Output) (M+M\_MVAL2)

## M+M\_MVAL2

This FB controls an ON/OFF motor valve (with intermediate status).

### FBD/LD



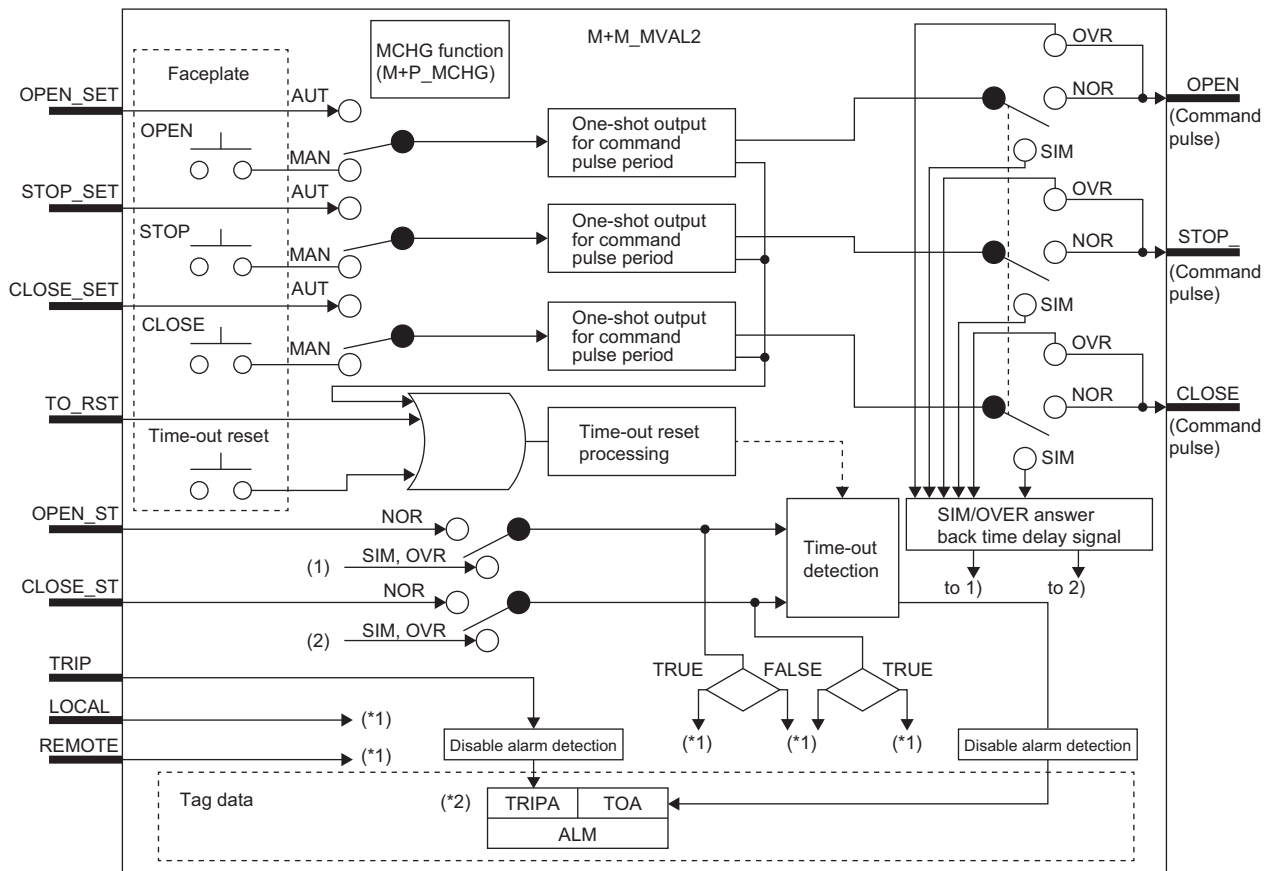
### Applicable tag type

MVAL2

### Control mode

MAN	AUT	CAS	CMV	CSV
○	○	—	—	—

### Block diagram



\*1 For displaying the faceplate status

\*2 Indicates bit item

## Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
OPEN_SET	External input for OPEN operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: OPEN	Input variable	BOOL
STOP_SET	External input for STOP operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: STOP	Input variable	BOOL
CLOSE_SET	External input for CLOSE operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: CLOSE	Input variable	BOOL
TO_RST	Time-out error external reset input	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: Time-out reset	Input variable	BOOL
OPEN_ST	Status answer input	TRUE: OPEN FALSE: SEMI_CLOSE* <sup>1</sup>	Input variable	BOOL
CLOSE_ST	Status answer input	TRUE: CLOSE FALSE: SEMI_CLOSE* <sup>1</sup>	Input variable	BOOL
TRIP	External failure (TRIP) input	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL
LOCAL	Local operation selection signal	TRUE: Valid FALSE: Invalid* <sup>1</sup>	Input variable	BOOL
REMOTE	Remote operation selection signal	TRUE: Valid FALSE: Invalid* <sup>1</sup>	Input variable	BOOL
OPEN	ON output for command pulse period	TRUE: Run FALSE: —* <sup>1</sup>	Output variable	BOOL
STOP_	ON output for command pulse period	TRUE: Run FALSE: —* <sup>1</sup>	Output variable	BOOL
CLOSE	ON output for command pulse period	TRUE: Run FALSE: —* <sup>1</sup>	Output variable	BOOL

\*1 The default value is FALSE.

### ■Public variable (others) \*<sup>1</sup>

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1, 2 1: MAN 2: AUT	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

### ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

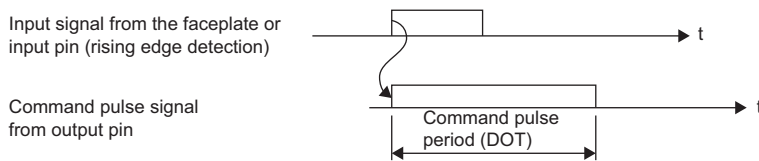
📄 Page 885 MVAL2

## Processing details

### ■ One-shot output for command pulse period

This function block performs one-shot output for command pulse period according to the operation from the faceplate or the input from the input variables (OPEN\_SET, STOP\_SET, CLOSE\_SET).

- When the setting is performed with the faceplate or the value of the input variable (OPEN\_SET) is changed from FALSE to TRUE, this function block outputs a command pulse signal (TRUE) from the output variable OPEN by the time period set with the command pulse period (DOT).
- When the setting is performed with the faceplate or the value of the input variable (STOP\_SET) is changed from FALSE to TRUE, this function block outputs a command pulse signal (TRUE) from the output variable STOP\_ by the time period set with the command pulse period (DOT).
- When the setting is performed with the faceplate or the value of the input variable (CLOSE\_SET) is changed from FALSE to TRUE, this function block outputs a command pulse signal (TRUE) from the output variable CLOSE by the time period set with the command pulse period (DOT).



### ■ Time-out detection/time-out reset

- Time-out detection

A time-out (TOA) of an alarm (ALM) will occur if TRUE is not input from the status answer input (OPEN\_ST/CLOSE\_ST) for a time longer than the time set with the time-out timer (TOT) after a command pulse signal (TRUE) is output from the output variable OPEN or CLOSE.

Condition	Alarm
	Time-out (TOA)
Time to a status answer signal input $\geq$ Time set with the time-out timer (TOT)	TRUE (Detected)
Time to a status answer signal input $<$ Time set with the time-out timer (TOT)	FALSE (Reset)

- Time-out reset

A time-out (TOA) of an alarm (ALM) is reset (FALSE) by either of the following operations.

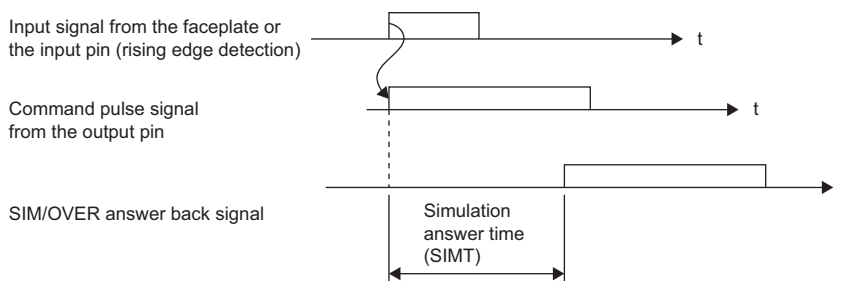
A command pulse signal is output from the output variable (OPEN, STOP\_, CLOSE) by the operation with the faceplate or the input from the input variable (OPEN\_SET, STOP\_SET, CLOSE\_SET).

TRUE is input to the input variable (TO\_RST).

### ■ SIM/OVER answer back time delay signal

In SIMULATION mode or OVERRIDE mode, this function block creates a status answer signal after a command signal output in the CPU module.

The lag time of this status answer signal is set with the simulation answer time (SIMT).





## ■Disable alarm detection

This function block sets whether to detect alarms (ALM) or not.

Item	Disable alarm detection processing
Disable alarm detection by setting tag data	If the following bits in the disable alarm detection (INH) are TRUE, no TRIPA and TOA of the alarm (ALM) will be detected. <ul style="list-style-type: none"> <li>• ERRI</li> <li>• TRIP1</li> <li>• TOI</li> </ul>

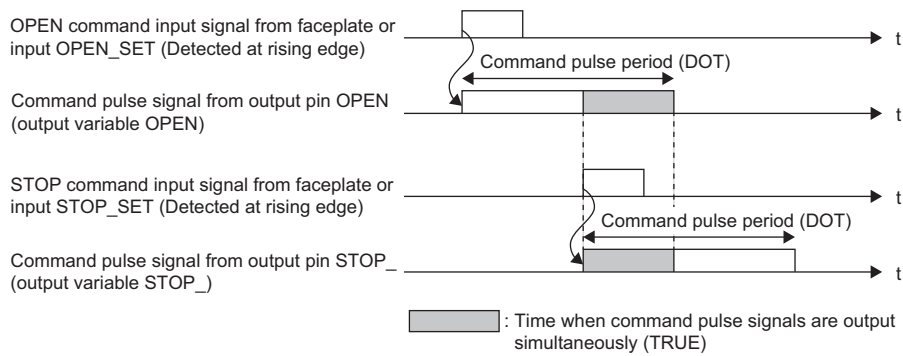
## Operation error

There is no operation error.



If another command is issued while a command pulse signal (TRUE) is output from the output variable OPEN, STOP\_, or CLOSE, the command pulse signals (TRUE) are output simultaneously.

<When multiple command pulse signals (TRUE) are output>



When the output variable OPEN, STOP\_, or CLOSE are output directly to the external device, multiple commands (OPEN command, STOP command, and CLOSE command) may be output simultaneously to the external device.

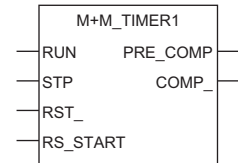
Not to output multiple commands to the external device simultaneously, create a program to output only either one of the command pulse signals to the external device.

# 18.5 Timer 1 (Timer stops when COMPLETE flag is ON) (M+M\_TIMER1)

## M+M\_TIMER1

This function block is a clock timer. It stops counting the time when the timer count value reaches the set value.

### FBD/LD



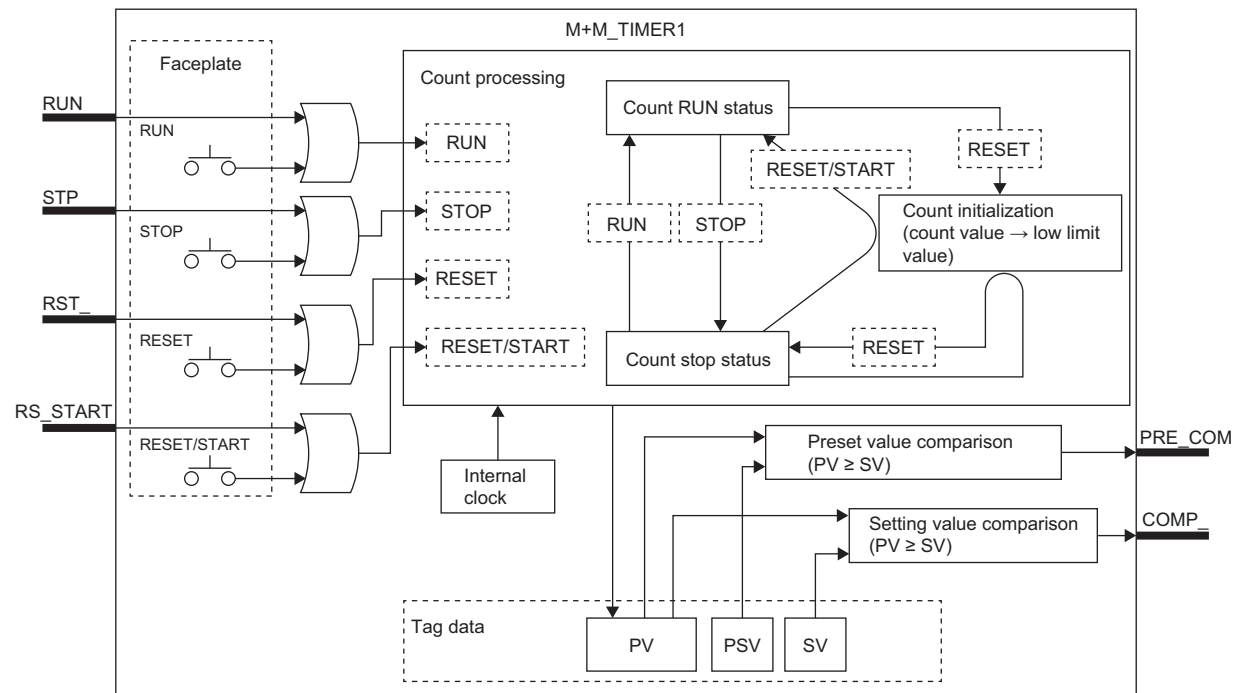
### Applicable tag type

TIMER1

### Control mode

MAN	AUT	CAS	CMV	CSV
—	—	—	—	—

### Block diagram



## Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
RUN	External input for RUN operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: RUN	Input variable	BOOL
STP	External input for STOP operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: STOP	Input variable	BOOL
RST_	External input for RESET operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: RESET	Input variable	BOOL
RS_START	External input for RESET/START operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: RESET/START	Input variable	BOOL
PRE_COMP	Preset value count up completed	TRUE: Complete FALSE: Not completed* <sup>1</sup>	Output variable	BOOL
COMP_	Setting value count up completed	TRUE: Complete FALSE: Not completed* <sup>1</sup>	Output variable	BOOL

\*1 The default value is FALSE.

### Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

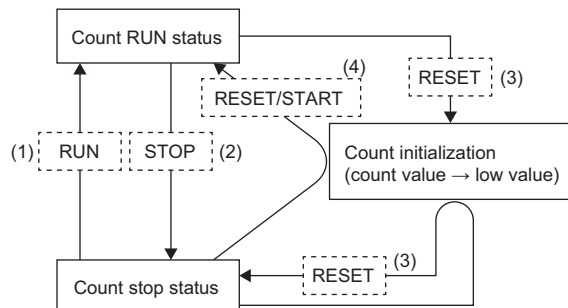
📖 Page 886 TIMER1

Set the following values in "FB Property".

- Estimated value (0 to 9999)
- Setting value (0 to 9999)
- Timer multiplying factor (second and minute)
- Timer high limit (0 to 9999)
- Timer low limit (0 to 9999)

## Processing details

### Count processing



- When the setting is performed with the faceplate or the value of the input variable (RUN) is changed from FALSE to TRUE, this function block stores the timer current value in the process variable (PV) in the unit set with the timer multiplying factor (MULT).  
When the process variable (PV) reaches the estimated value (PSV), TRUE will be output from the output variable PRE\_COMP. (Preset value comparison in the block diagram)  
When the process variable (PV) reaches the setting value (SV), TRUE will be output from the output variable COMP\_, the timer stops, and the status of the count processing becomes the count stop status. (Setting value comparison in the block diagram)  
When the process variable (PV) reaches the timer high limit (RH), the timer stops.
- When the setting is performed with the faceplate or the value of the input variable (STP) is changed from FALSE to TRUE, measuring the process variable (PV) will be stopped.
- When the setting is performed with the faceplate or the value of the input variable (RST\_) is changed from FALSE to TRUE, the process variable (PV) will be set to the timer lower limit (RL) value, and the timer stops.
- When the setting is performed with the faceplate or the value of the input variable (RS\_START) is changed from FALSE to TRUE, the process variable (PV) will be set to the timer low limit (RL) value, and the timer starts.

## Operation error

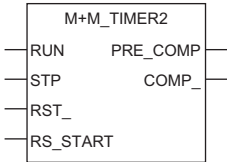
There is no operation error.

# 18.6 Timer 2 (Timer continues when COMPLETE flag is ON) (M+M\_TIMER2)

## M+M\_TIMER2

This function block is a clock timer. It continues to count the time even if the time count value exceeds the set value, and the timer stops when the time count value reaches the timer high limit value.

### FBD/LD



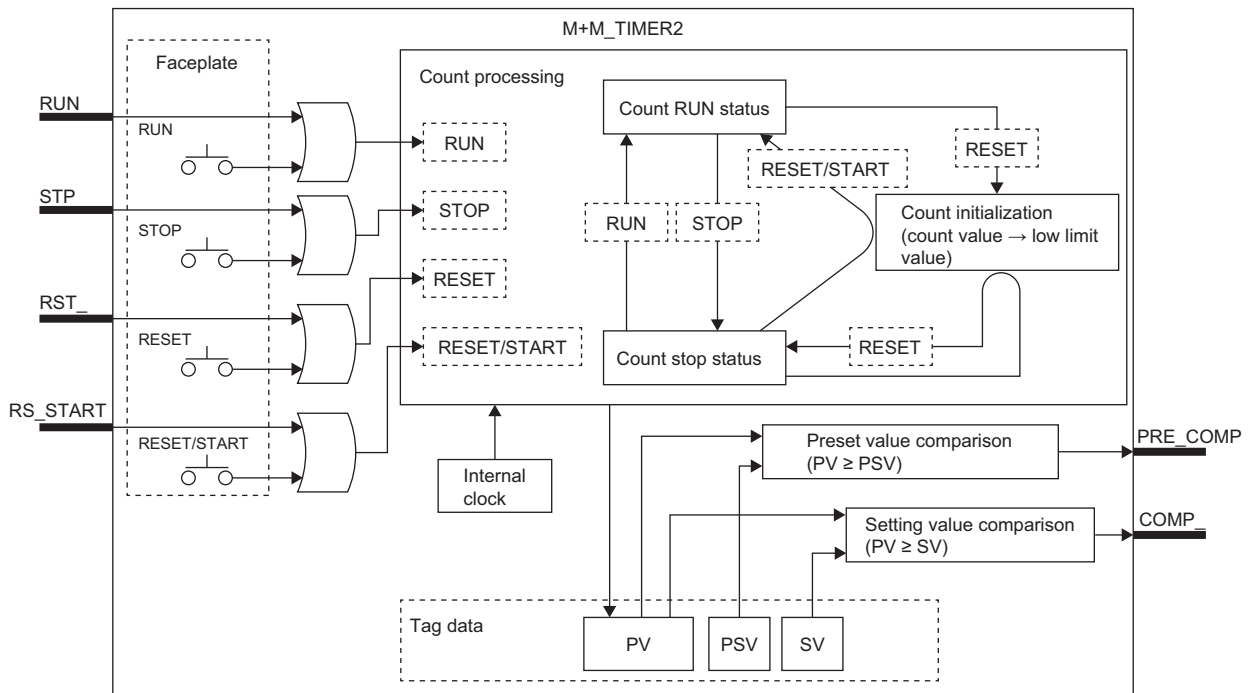
### Applicable tag type

TIMER2

### Control mode

MAN	AUT	CAS	CMV	CSV
—	—	—	—	—

### Block diagram



## Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
RUN	External input for RUN operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: RUN	Input variable	BOOL
STP	External input for STOP operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: STOP	Input variable	BOOL
RST_	External input for RESET operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: RESET	Input variable	BOOL
RS_START	External input for RESET/START operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: RESET/START	Input variable	BOOL
PRE_COMP	Preset value count up completed	TRUE: Complete FALSE: Not completed* <sup>1</sup>	Output variable	BOOL
COMP_	Setting value count up completed	TRUE: Complete FALSE: Not completed* <sup>1</sup>	Output variable	BOOL

\*1 The default value is FALSE.

### Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

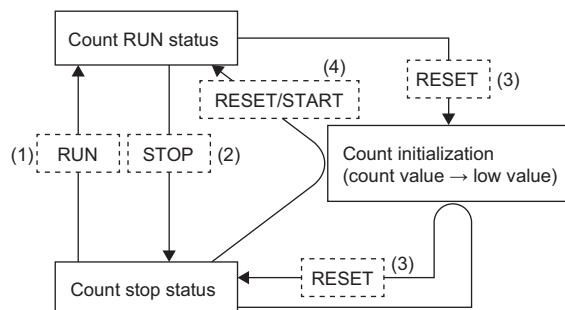
📖 Page 887 TIMER2

Set the following values in "FB Property".

- Estimated value (0 to 9999)
- Setting value (0 to 9999)
- Timer multiplying factor (second and minute)
- Timer high limit (0 to 9999)
- Timer low limit (0 to 9999)

## Processing details

### Count processing



- (1) When the setting is performed with the faceplate or the value of the input variable (RUN) is changed from FALSE to TRUE, this function block stores the timer current value in the process variable (PV) in the unit set with the timer multiplying factor (MULT).  
When the process variable (PV) reaches the estimated value (PSV), TRUE will be output from the output variable PRE\_COMP. (Preset value comparison in the block diagram)  
When the process variable (PV) reaches the setting value (SV), TRUE will be output from the output variable COMP\_. (Setting value comparison in the block diagram)  
When the process variable (PV) reaches the timer high limit (RH), the timer stops.
- (2) When the setting is performed with the faceplate or the value of the input variable (STP) is changed from FALSE to TRUE, measuring the process variable (PV) will be stopped.
- (3) When the setting is performed with the faceplate or the value of the input variable (RST\_) is changed from FALSE to TRUE, the process variable (PV) will be set to the timer lower limit (RL) value, and the timer stops.
- (4) When the setting is performed with the faceplate or the value of the input variable (RS\_START) is changed from FALSE to TRUE, the process variable (PV) will be set to the timer low limit (RL) value, and the timer starts.

## Operation error

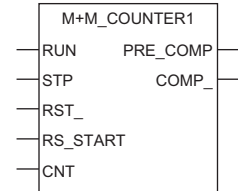
There is no operation error.

# 18.7 Counter 1 (Counter stops when COMPLETE flag is ON) (M+M\_COUNTER1)

## M+M\_COUNTER1

This function block is a counter that counts contact signal inputs. It stops counting the inputs when the count value reaches the set value.

### FBD/LD



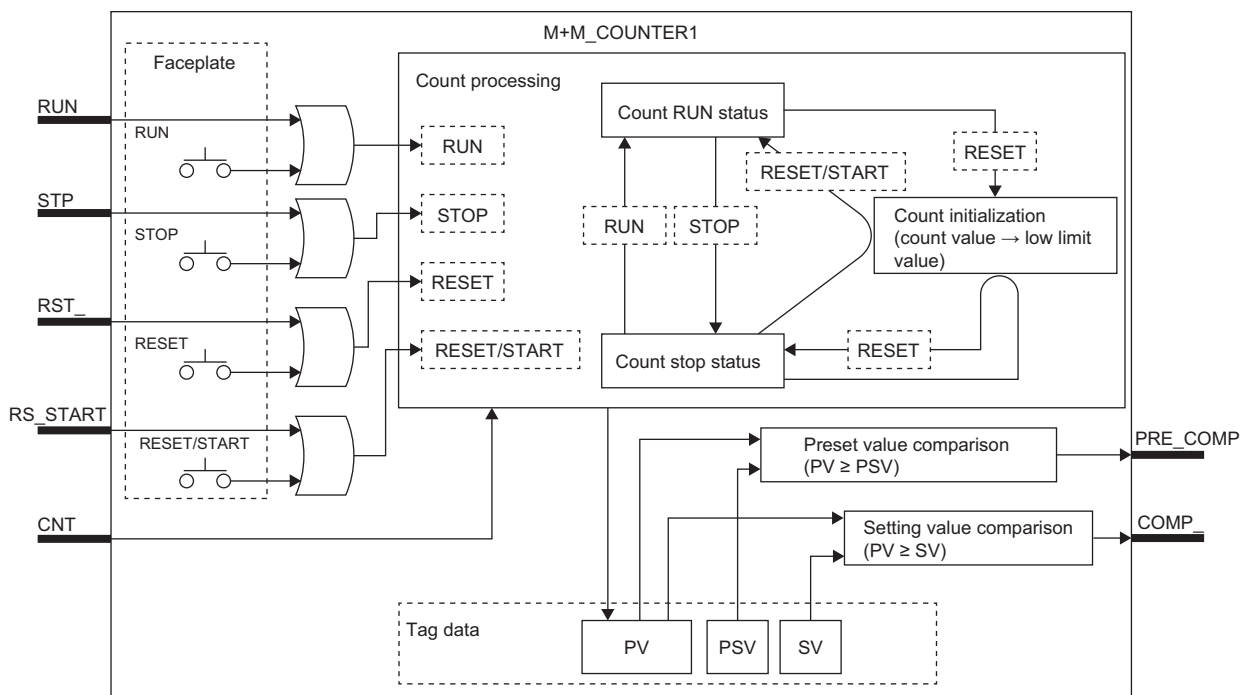
### Applicable tag type

COUNT1

### Control mode

MAN	AUT	CAS	CMV	CSV
—	—	—	—	—

### Block diagram



## Setting data


### Input/output variable

Variable name	Description	Recommended range	Type	Data type
RUN	External input for RUN operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: RUN	Input variable	BOOL
STP	External input for STOP operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: STOP	Input variable	BOOL
RST_val	External input for RESET operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: RESET	Input variable	BOOL
RS_START	External input for RESET/START operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: RESET/START	Input variable	BOOL
CNT	Count contact signal input	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: Count	Input variable	BOOL
PRE_COMP	Preset value count up completed	TRUE: Complete FALSE: Not completed* <sup>1</sup>	Output variable	BOOL
COMP_val	Setting value count up completed	TRUE: Complete FALSE: Not completed* <sup>1</sup>	Output variable	BOOL

\*1 The default value is FALSE.

### Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

 Page 888 COUNT1

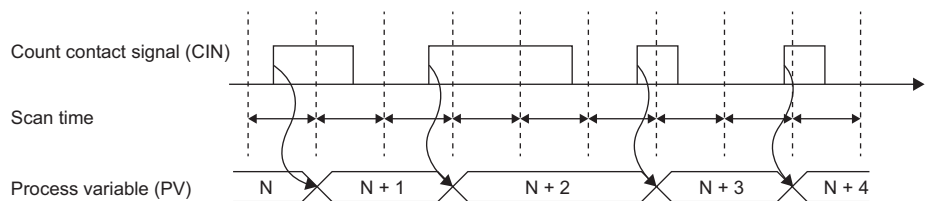
Set the following values in "FB Property".

- Estimated value (0 to 9999)
- Setting value (0 to 9999)
- Timer multiplying factor (second and minute)
- Timer high limit (0 to 9999)
- Timer low limit (0 to 9999)

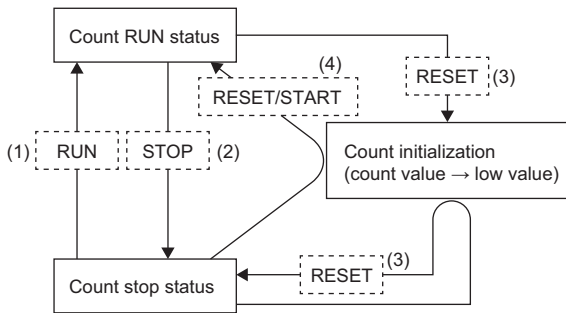
## Processing details

### Count input (CNT)

This function block detects the rising edges (FALSE → TRUE) of the input variable (CIN) and performs count processing (current value +1).



## Count processing



- (1) When the setting is performed with the faceplate or the value of the input variable (RUN) is changed from FALSE to TRUE, this function block stores the current value + 1 (counter value) in the process variable (PV) in the unit set with the counter multiplying factor (MULT).  
When the process variable (PV) reaches the estimated value (PSV), TRUE will be output from the output variable PRE\_COMP. (Preset value comparison in the block diagram)  
When the process variable (PV) reaches the setting value (SV), TRUE will be output from the output variable COMP\_, the counter stops, and the status of the count processing becomes the count stop status. (Setting value comparison in the block diagram)  
When the process variable (PV) reaches the counter high limit (RH), the counter stops.
- (2) When the setting is performed with the faceplate or the value of the input variable (STP) is changed from FALSE to TRUE, measuring the process variable (PV) will be stopped.
- (3) When the setting is performed with the faceplate or the value of the input variable (RST\_) is changed from FALSE to TRUE, the process variable (PV) will be set to the counter low limit (RL) value, and the counter stops.
- (4) When the setting is performed with the faceplate or the value of the input variable (RS\_START) is changed from FALSE to TRUE, the process variable (PV) will be set to the counter low limit (RL) value, and the counter starts.

### Operation error

There is no operation error.

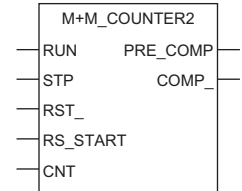


# 18.8 Counter 2 (Counter continues when COMPLETE flag is ON) (M+M\_COUNTER2)

## M+M\_COUNTER2

This function block is a counter that counts contact signal inputs. It continues to count the inputs even if the count value exceeds the set value, and stops counting the input when the count value reaches the counter high limit value.

### FBD/LD



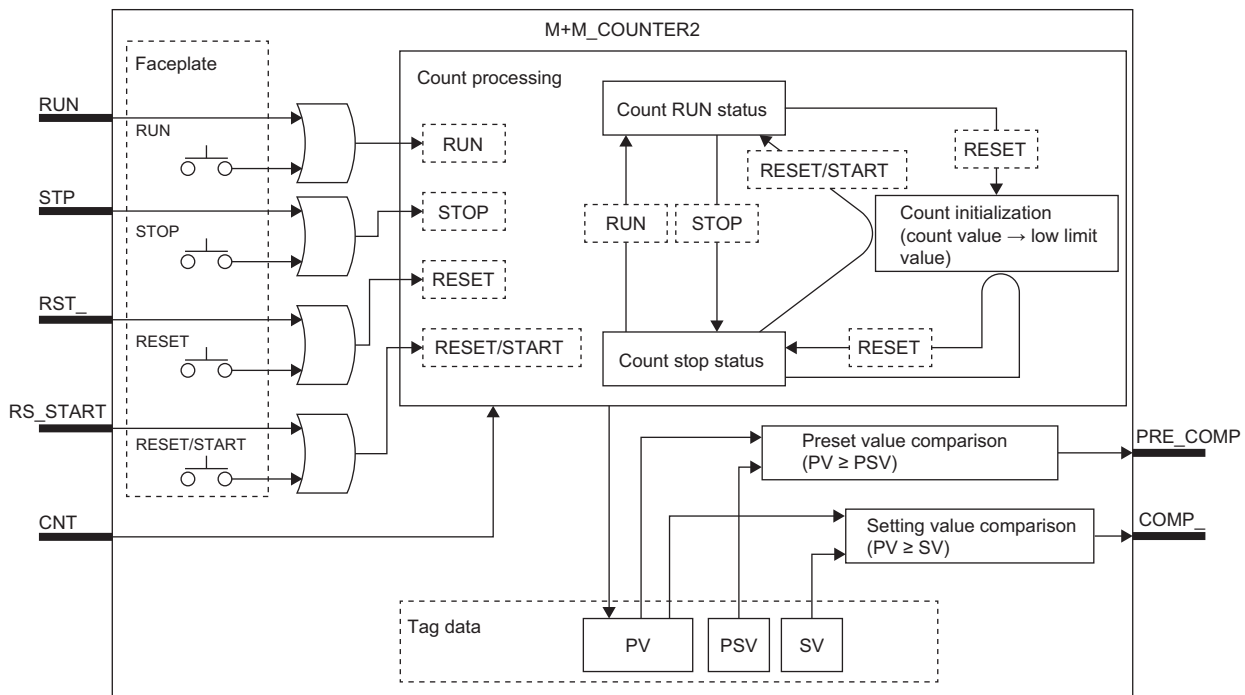
### Applicable tag type

COUNT2

### Control mode

MAN	AUT	CAS	CMV	CSV
—	—	—	—	—

### Block diagram



## Setting data


### Input/output variable

Variable name	Description	Recommended range	Type	Data type
RUN	External input for RUN operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: RUN	Input variable	BOOL
STP	External input for STOP operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: STOP	Input variable	BOOL
RST_	External input for RESET operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: RESET	Input variable	BOOL
RS_START	External input for RESET/START operation	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: RESET/START	Input variable	BOOL
CNT	Count contact signal input	TRUE, FALSE* <sup>1</sup> FALSE → TRUE: Count	Input variable	BOOL
PRE_COMP	Preset value count up completed	TRUE: Complete FALSE: Not completed* <sup>1</sup>	Output variable	BOOL
COMP_	Setting value count up completed	TRUE: Complete FALSE: Not completed* <sup>1</sup>	Output variable	BOOL

\*1 The default value is FALSE.

### Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

 Page 889 COUNT2

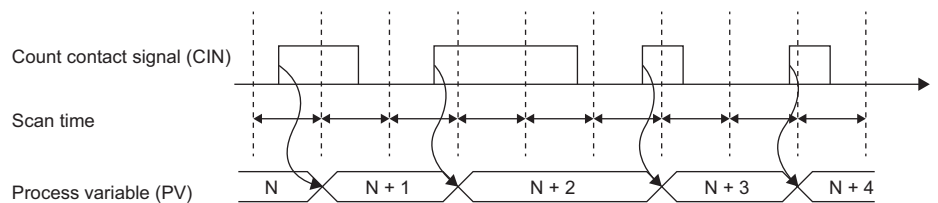
Set the following values in "FB Property".

- Estimated value (0 to 9999)
- Setting value (0 to 9999)
- Timer multiplying factor (second and minute)
- Timer high limit (0 to 9999)
- Timer low limit (0 to 9999)

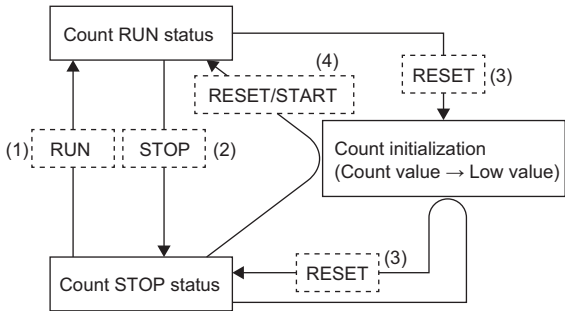
## Processing details

### Count input (CNT)

This function block detects the rising edges (FALSE → TRUE) of the input variable (CIN) and performs count processing (current value +1).



## Count processing



- (1) When the setting is performed with the faceplate or the value of the input variable (RUN) is changed from FALSE to TRUE, this function block stores the current value + 1 (counter value) in the process variable (PV) in the unit set with the counter multiplying factor (MULT).  
When the process variable (PV) reaches the estimated value (PSV), TRUE will be output from the output variable PRE\_COMP. (Preset value comparison in the block diagram)  
When the process variable (PV) reaches the setting value (SV), TRUE will be output from the output variable COMP\_-. (Setting value comparison in the block diagram)  
When the process variable (PV) reaches the counter high limit (RH), the counter stops.
- (2) When the setting is performed with the faceplate or the value of the input variable (STP) is changed from FALSE to TRUE, measuring the process variable (PV) will be stopped.
- (3) When the setting is performed with the faceplate or the value of the input variable (RST\_) is changed from FALSE to TRUE, the process variable (PV) will be set to the counter low limit (RL) value, and the counter stops.
- (4) When the setting is performed with the faceplate or the value of the input variable (RS\_START) is changed from FALSE to TRUE, the process variable (PV) will be set to the counter low limit (RL) value, and the counter starts.

### Operation error

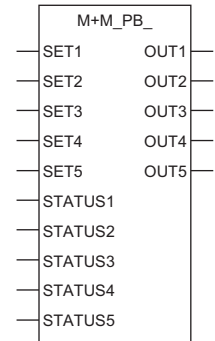
There is no operation error.

# 18.9 Push Button Operation (5 Input/5 Output) (M+M\_PB\_)

## M+M\_PB\_

This FB performs push button operation.

### FBD/LD



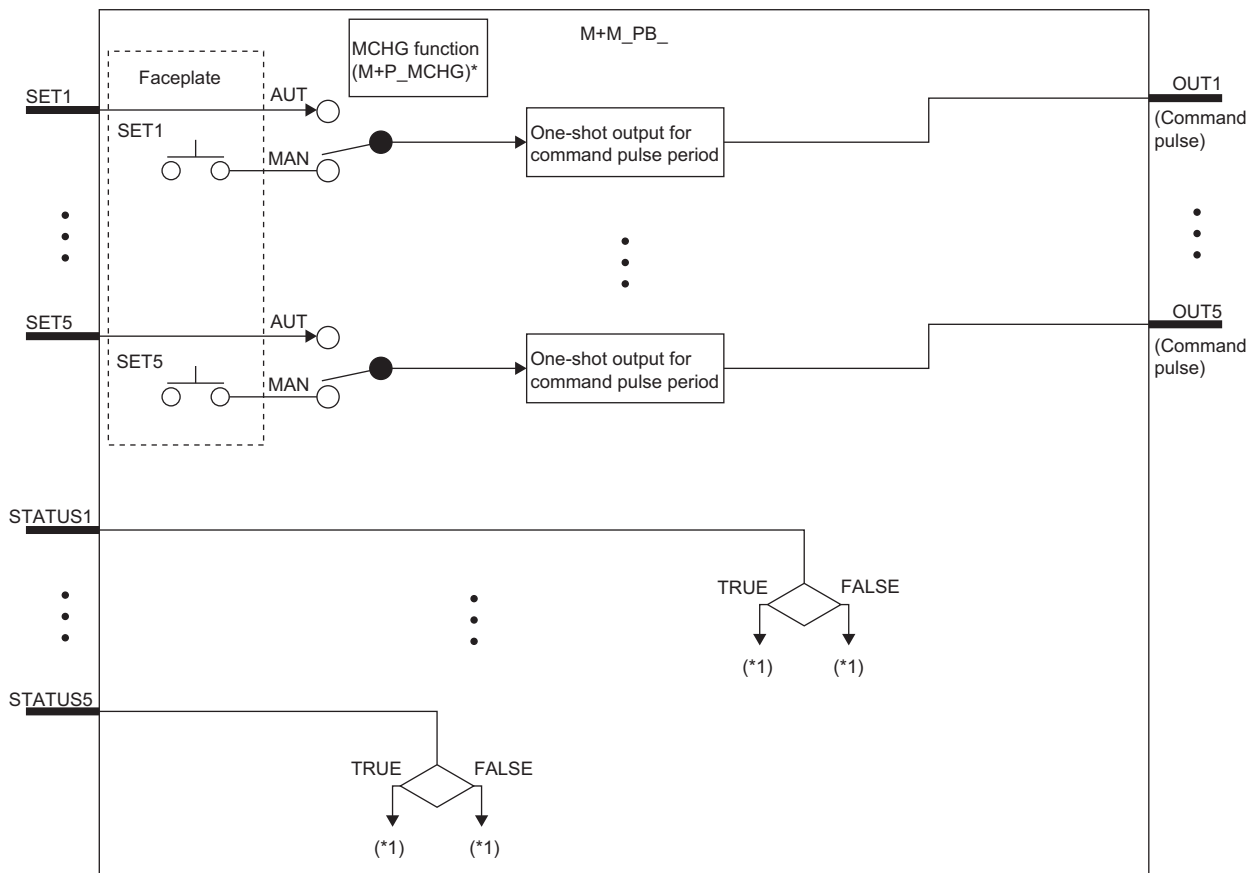
### Applicable tag type

PB

### Control mode

MAN	AUT	CAS	CMV	CSV
○	○	—	—	—

### Block diagram



\*1 For displaying the faceplate status.

## Setting data

### Input/output variable

Variable name	Description	Recommended range	Type	Data type
SET1	External input of OUT1 operation	TRUE, FALSE FALSE → TRUE: ON	Input variable	BOOL
SET2	External input of OUT2 operation	TRUE, FALSE FALSE → TRUE: ON	Input variable	BOOL
SET3	External input of OUT3 operation	TRUE, FALSE FALSE → TRUE: ON	Input variable	BOOL
SET4	External input of OUT4 operation	TRUE, FALSE FALSE → TRUE: ON	Input variable	BOOL
SET5	External input of OUT5 operation	TRUE, FALSE FALSE → TRUE: ON	Input variable	BOOL
STATUS1	Status 1 answer input	TRUE: ON FALSE: OFF	Input variable	BOOL
STATUS2	Status 2 answer input	TRUE: ON FALSE: OFF	Input variable	BOOL
STATUS3	Status 3 answer input	TRUE: ON FALSE: OFF	Input variable	BOOL
STATUS4	Status 4 answer input	TRUE: ON FALSE: OFF	Input variable	BOOL
STATUS5	Status 5 answer input	TRUE: ON FALSE: OFF	Input variable	BOOL
OUT1	ON output for command pulse period	TRUE: Command FALSE: —	Output variable	BOOL
OUT2	ON output for command pulse period	TRUE: Command FALSE: —	Output variable	BOOL
OUT3	ON output for command pulse period	TRUE: Command FALSE: —	Output variable	BOOL
OUT4	ON output for command pulse period	TRUE: Command FALSE: —	Output variable	BOOL
OUT5	ON output for command pulse period	TRUE: Command FALSE: —	Output variable	BOOL

### Public variable (others) \*1

- Control mode change processing (MCHG function)

Variable name	Description	Recommended range	Initial value	Set by	Data type
MODEIN	Mode Change Signal	1: MAN 2: AUT	0	User	INT
E_	Enable Mode Change	TRUE: Execute FALSE: Stop	FALSE	User	BOOL

\*1 Read or write the variables using a program. They are not displayed on "FB Property" of the engineering tool.

### Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

 Page 890 PB

### Function of the components

This tag FB consists of the following function blocks.

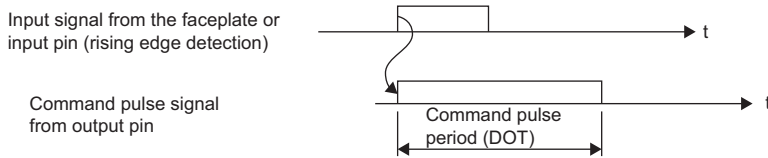
Item	FB	Reference
MCHG function	M+P_MCHG	Page 378 M+P_MCHG

## Processing details

### ■ One-shot output for command pulse period

This function block performs one-shot output for command pulse period according to the operation from the faceplate or the input from the input variable (SET1 to SET5).

- When the setting is performed with the faceplate or the value of the input variable (SET1 to SET5) is changed from FALSE to TRUE, this function block outputs a command pulse signal (TRUE) from the output variable (OUT1 to OUT5) by the time period set with the command pulse period (DOT).

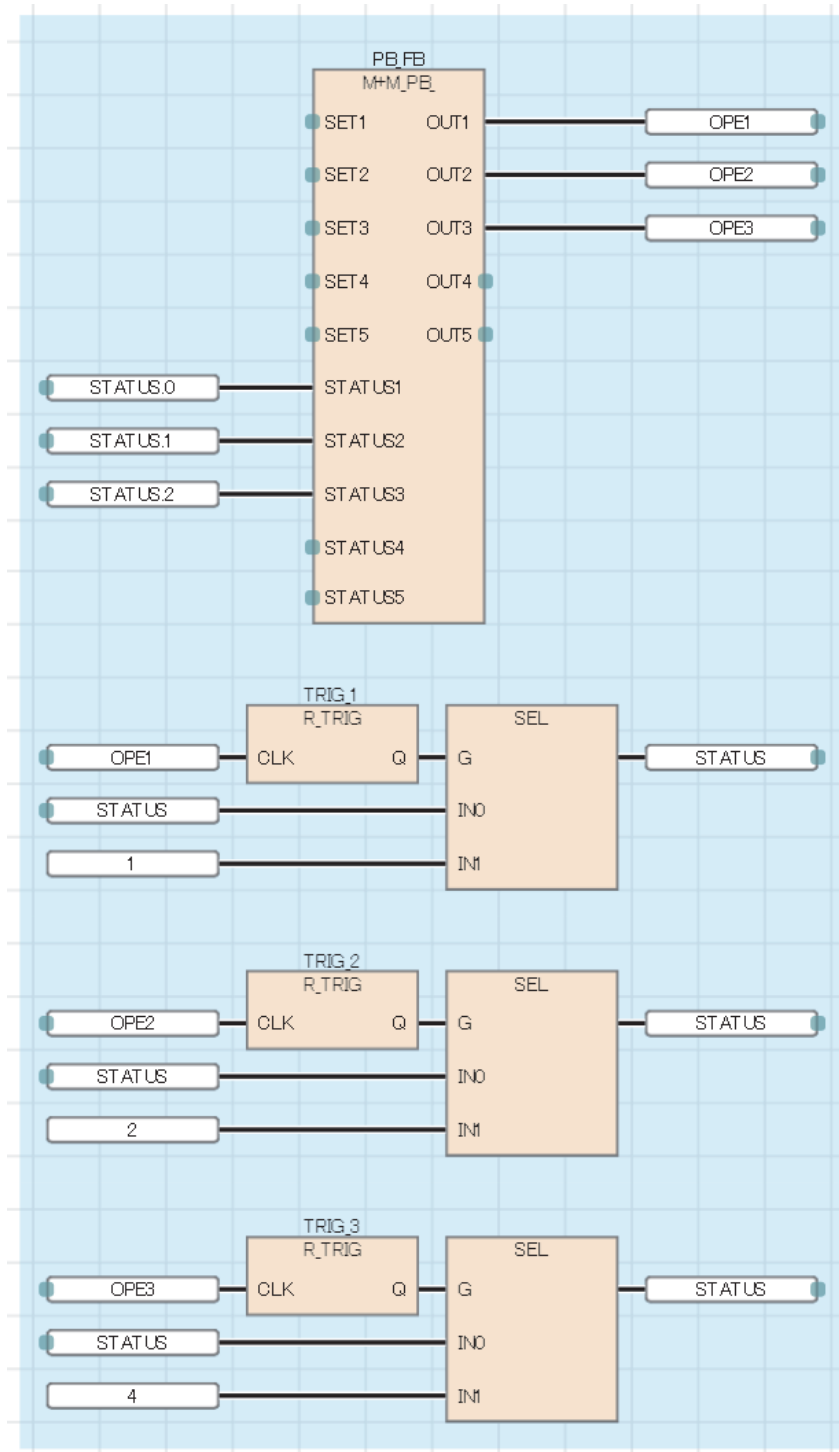


## Operation error

There is no operation error.

## Program example

Configuring a radio button with three buttons



### Point

On the faceplate, a one-shot command can be output by clicking each button, and the name of the ON/OFF state can be displayed.

In addition, combinations of the buttons enable operations similar to radio buttons.

# 19 ALARM TAG

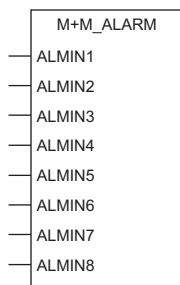
The following FBs perform alarm notification.

## 19.1 Alarm (M+M\_ALARM)

### M+M\_ALARM

This function block displays the alarms for the input pins ALMIN1 to ALMIN8 to which TRUE is input on the "Alarm List" window of the PX Developer monitor tool.

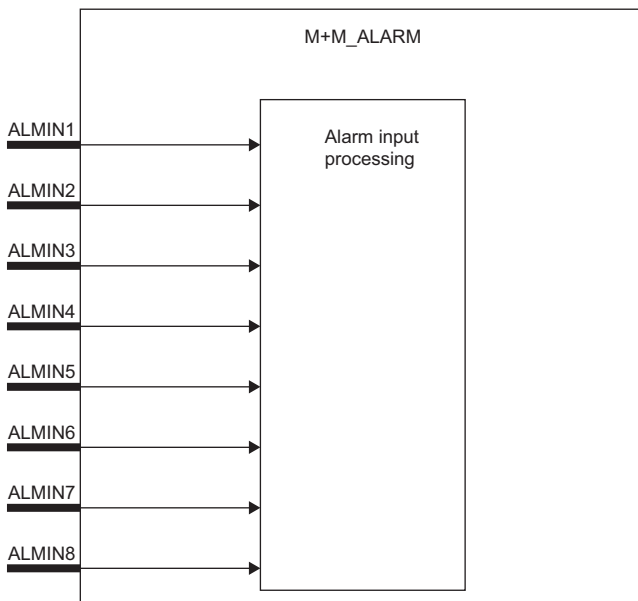
#### FBD/LD



#### Applicable tag type

ALM

#### ■ Block diagram





## Setting data

### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
ALMIN1	Alarm 1 input signal	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL
ALMIN2	Alarm 2 input signal	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL
ALMIN3	Alarm 3 input signal	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL
ALMIN4	Alarm 4 input signal	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL
ALMIN5	Alarm 5 input signal	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL
ALMIN6	Alarm 6 input signal	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL
ALMIN7	Alarm 7 input signal	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL
ALMIN8	Alarm 8 input signal	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL

\*1 The default value is FALSE.

### ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

 Page 891 ALM

## Processing details

### ■Alarm input processing

This function block displays the alarms for the input pins ALMIN1 to ALMIN8 to which TRUE is input on the "Alarm List" window of the PX Developer monitor tool.

For the operating method of PX Developer monitor tool, refer to the following.

 PX Developer Version 1 Operating Manual (Monitor Tool)

## Operation error

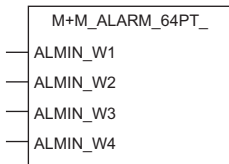
There is no operation error.

# 19.2 64 Points Alarm (M+M\_ALARM\_64PT\_)

## M+M\_ALARM\_64PT\_

This function block displays the alarms for the bits of the input pins ALMIN\_W1 to ALMIN\_W4 to which TRUE is input on the "Alarm List" window of the PX Developer monitor tool.

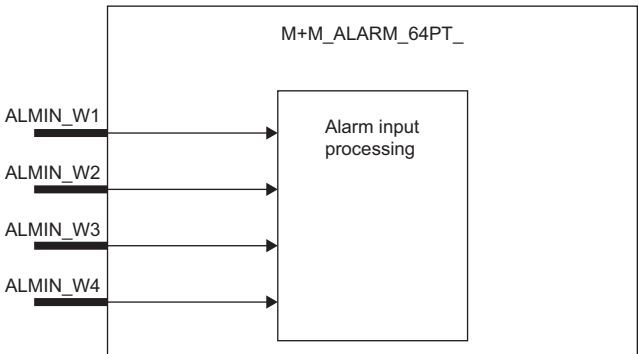
### FBD/LD



### Applicable tag type

ALM\_64PT

### Block diagram



### Setting data

#### Input/output variable

Variable name	Description	Recommended range	Type	Data type
ALMIN_W1	Alarm 1 to 16 input signal The bits correspond to the alarms 1 to 16 starting from bit 0. (For each bit, TRUE: Detected, FALSE: Reset)	0H to FFFFH	Input variable	WORD
ALMIN_W2	Alarm 17 to 32 input signal The bits correspond to the alarms 17 to 32 starting from bit 0. (For each bit, TRUE: Detected, FALSE: Reset)	0H to FFFFH	Input variable	WORD
ALMIN_W3	Alarm 33 to 48 input signal The bits correspond to the alarms 33 to 48 starting from bit 0. (For each bit, TRUE: Detected, FALSE: Reset)	0H to FFFFH	Input variable	WORD
ALMIN_W4	Alarm 49 to 64 input signal The bits correspond to the alarms 49 to 64 starting from bit 0. (For each bit, TRUE: Detected, FALSE: Reset)	0H to FFFFH	Input variable	WORD

#### Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

📖 Page 892 ALM\_64PT

### Processing details

#### Alarm input processing

This function block displays the alarms for the bits of the input pins ALMIN\_W1 to ALMIN\_W4 to which TRUE is input on the "Alarm List" window of the PX Developer monitor tool.

For the operating method of PX Developer monitor tool, refer to the following.

📖 PX Developer Version 1 Operating Manual (Monitor Tool)

## Operation error

There is no operation error.

# 20 MESSAGE TAG

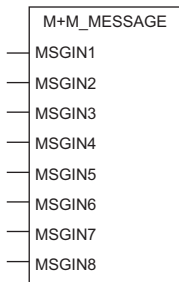
The following FBs perform message notification.

## 20.1 Message (M+M\_MESSAGE)

### M+M\_MESSAGE

This function block displays the events for the input pins MSGIN1 to MSGIN8 to which TRUE is input on the "Event List" window of the PX Developer monitor tool.

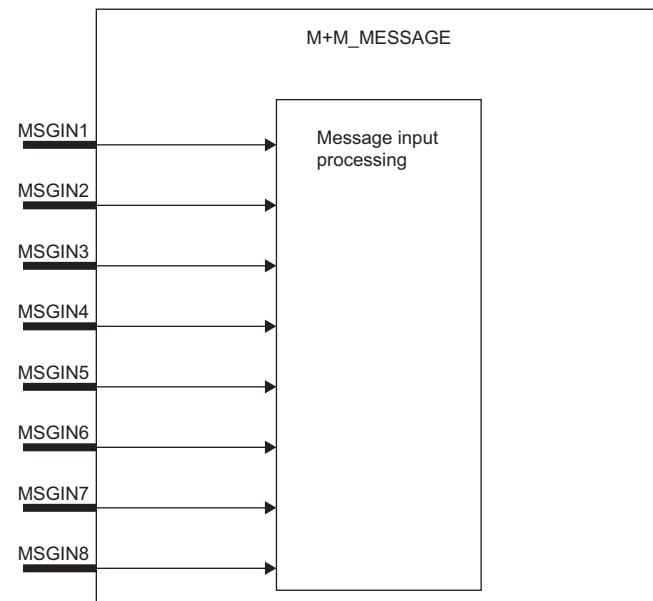
#### FBD/LD



#### Applicable tag type

MSG

#### ■ Block diagram



## Setting data


### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
MSGIN1	Message 1 input signal	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL
MSGIN2	Message 2 input signal	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL
MSGIN3	Message 3 input signal	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL
MSGIN4	Message 4 input signal	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL
MSGIN5	Message 5 input signal	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL
MSGIN6	Message 6 input signal	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL
MSGIN7	Message 7 input signal	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL
MSGIN8	Message 8 input signal	TRUE: Detected FALSE: Reset* <sup>1</sup>	Input variable	BOOL

\*1 The default value is FALSE.

### ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

 Page 899 MSG

## Processing details

### ■Message input processing

This function block displays the messages for the input pins MSGIN1 to MSGIN8 to which TRUE is input on the "Event List" window of the PX Developer monitor tool.

For the operating method of PX Developer monitor tool, refer to the following.

 PX Developer Version 1 Operating Manual (Monitor Tool)

## Operation error

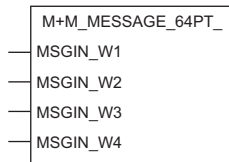
There is no operation error.

## 20.2 64 Points Message (M+M\_MESSAGE\_64PT\_)

### M+M\_MESSAGE\_64PT\_

This function block displays the events for the bits of the input pins MSGIN\_W1 to MSGIN\_W4 to which TRUE is input on the "Event List" window of the PX Developer monitor tool.

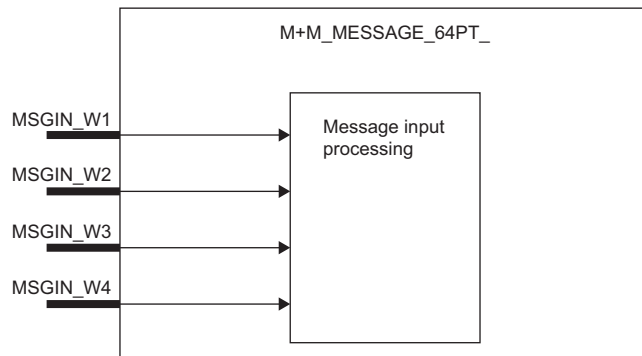
#### FBD/LD



#### Applicable tag type

MSG\_64PT

### ■Block diagram



### Setting data

#### ■Input/output variable

Variable name	Description	Recommended range	Type	Data type
MSGIN_W1	Message 1 to 16 input signal The bits correspond to the messages 1 to 16 starting from bit 0. (For each bit, TRUE: Detected, FALSE: Reset)	0H to FFFFH	Input variable	WORD
MSGIN_W2	Message 17 to 32 input signal The bits correspond to the messages 17 to 32 starting from bit 0. (For each bit, TRUE: Detected, FALSE: Reset)	0H to FFFFH	Input variable	WORD
MSGIN_W3	Message 33 to 48 input signal The bits correspond to the messages 33 to 48 starting from bit 0. (For each bit, TRUE: Detected, FALSE: Reset)	0H to FFFFH	Input variable	WORD
MSGIN_W4	Message 49 to 64 input signal The bits correspond to the messages 49 to 64 starting from bit 0. (For each bit, TRUE: Detected, FALSE: Reset)	0H to FFFFH	Input variable	WORD

#### ■Tag data

For details on tag data that is read or written with this tag FB, refer to the following.

📖 Page 900 MSG\_64PT

### Processing details

#### ■Message input processing

This function block displays the messages for the bits of the input pins MSGIN\_W1 to MSGIN\_W4 to which TRUE is input on the "Event List" window of the PX Developer monitor tool.

For the operating method of PX Developer monitor tool, refer to the following.

📖 PX Developer Version 1 Operating Manual (Monitor Tool)

## Operation error

There is no operation error.

# MEMO

---



# PART 6

# PROCESS CONTROL INSTRUCTIONS

This part consists of the following chapter.

21 PROCESS CONTROL INSTRUCTIONS

---

# 21 PROCESS CONTROL INSTRUCTIONS

## Point

When a process control program is created, using process control function blocks is recommended

Process control function blocks have features as follows.

- A process control program can be easily created by placing and connecting FB elements.
- Since the initial value of the function block can be set in the "FB Property" window of the engineering tool, the program for the initial value setting is not required.
- An operation constant can be input to a label indicating a tag name without being conscious of address of a device.
- The operating status of a tag FB can be checked and controlled by accessing the tag data from the faceplate of an engineering tool.

For the overview of process control function blocks, refer to the following.

☞ Page 20 PROCESS CONTROL FUNCTION BLOCKS AND PROCESS CONTROL INSTRUCTIONS

## 21.1 Overview

This section describes the loop type that can be configured by process control instructions, data configurations of the instructions, instruction execution methods, and precautions.

### Basic loop types

The following table summarizes basic loop types configured by combinations of process control instructions.

Loop type	Configuration	Application
Two-degree-of-freedom PID control (S2PID)		Used for general PID control (two degrees of freedom). (Speed type) PID operation is performed every control cycle.
PID control (SPID)		Used for general PID control. (Speed type) PID operation is performed every control cycle.
PIDP control (SPIDP)		Used for general PID control. (Position type) PID operation is performed every control cycle.
Sample PI control (SSPI)		Used for processes which involve much dead time. The operation to perform PI control for the operating time which is set and stabilize the output after that is repeated every sample cycle.
I-PD control (SIPD)		Used to make a slow response so as not to give a shock to the operation terminal and process when the set value is changed.

Loop type	Configuration	Application
Blend PI control (SBPI)		Used for processes which allow a constant manipulated value in the long run even if it varies in the short term.
Ratio control (SR)		Performs control so that a given manipulated value keeps a constant ratio with other rates of change.
Two-position (on/off) control (SONF2)		Performs control so that the manipulated value is turned on or off depending on whether the deviation is positive or negative.
Three-position (on/off) control (SONF3)		Performs control by outputting three-area signals for process values. This control can suppress rapid changes in the manipulated value.
Batch counter (SBC)		Performs valve on/off control in the process of batch charging to the tank.
Program setter (SPGS)		Outputs according to the temporal change of the value that has been set in advance.
Manual output (SMOUT)		Operates the operation terminal for manual output.
Monitor (SMON)		Inputs process values to detect process errors such as upper/lower limit alarms.
Manual output with monitor (SMWM)		Performs manual operation while inputting process values to check that no error is caused.
Selector (SSEL)		Used for signal selection.

### Point

For the loop type processing time, refer to the following.

MELSEC iQ-R CPU Module User's Manual (Application)

# Process control instructions and data configuration

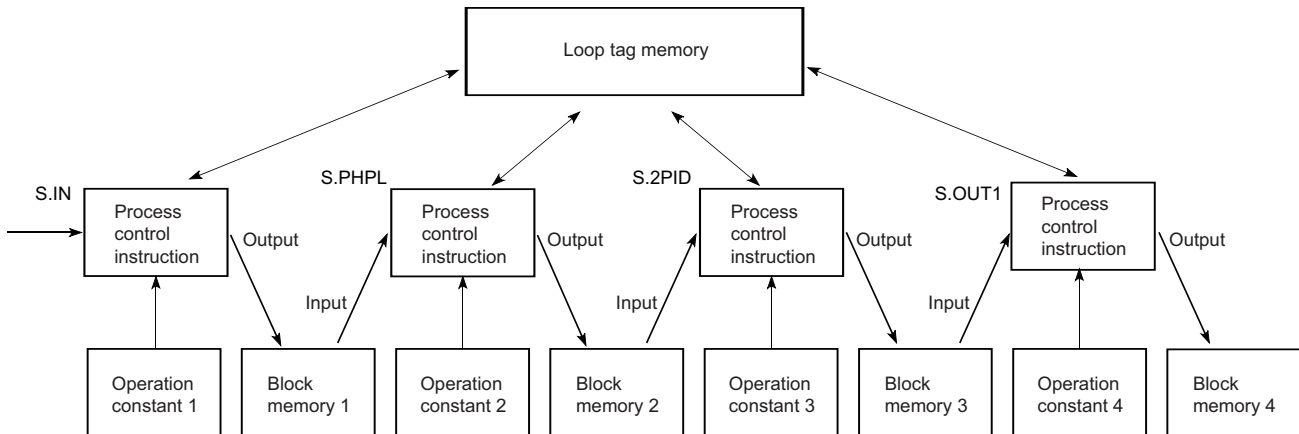
This section describes the data configurations used by process control instructions.

## Data configuration in which loop tags are used

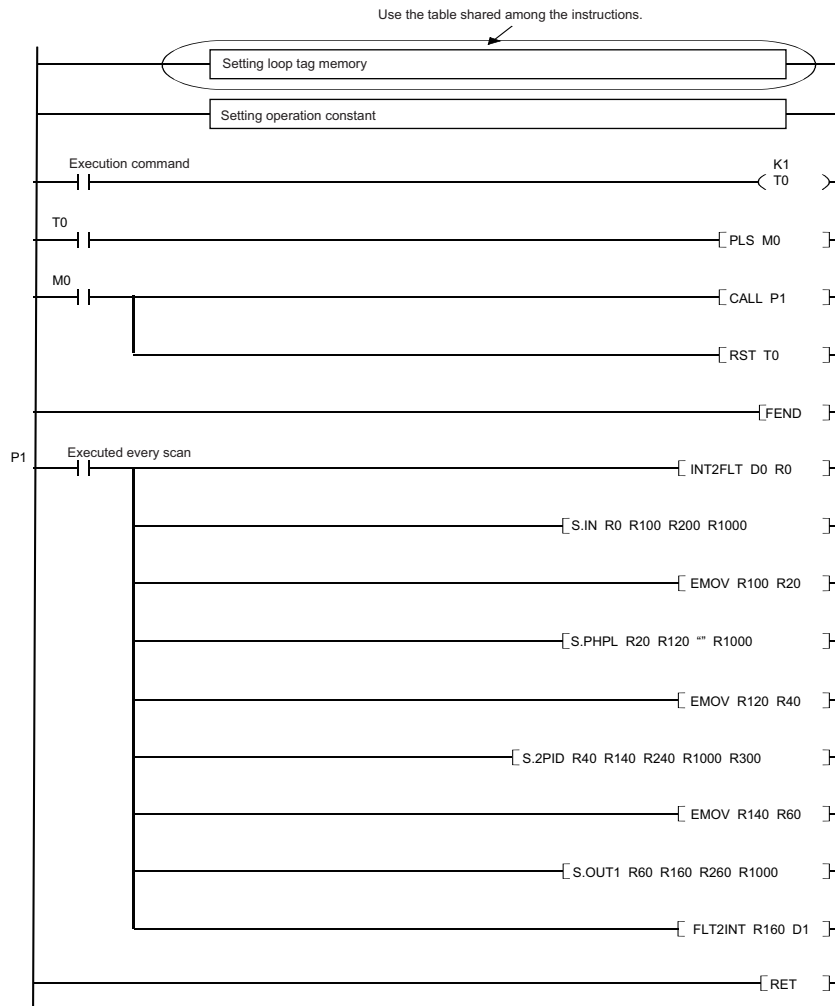
The process control instructions use control information common to each loop by storing it in common memory. The group of this common information is called a loop tag, and the storage memory is called loop tag memory. Loop monitoring and tuning can be performed by monitoring the loop tag.

**Ex.**

Block diagram of two-degree-of-freedom PID control (S2PID)



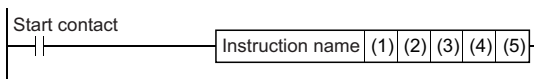
# Positioning the loop tag memory and operation constants based on the ladder diagram



• Loop tag memory setting

Loop tag memory (96 words)	Instruction used	Item	Standard value setting	Data type
+0	—	—	—	16-bit unsigned binary
+1	—	MODE	8H	16-bit unsigned binary
+3	—	ALM	4000H	16-bit unsigned binary
+4	—	INH	0H	16-bit unsigned binary
+10	S.PHPL	PV	0.0	Single-precision real number
+12	S.OUT1	MV	0.0	Single-precision real number
+14	S.2PID	SV	0.0	Single-precision real number
+16	S.2PID	DV	0.0	Single-precision real number
+18	S.OUT1	MH	100.0	Single-precision real number
+20	S.OUT1	ML	0.0	Single-precision real number
+22	S.PHPL	RH	100.0	Single-precision real number
⋮	⋮	⋮	⋮	⋮
+46	S.2PID	CT	1.0	Single-precision real number
+48	S.OUT1	DML	100.0	Single-precision real number
+50	S.2PID	DVL	100.0	Single-precision real number
+52	S.2PID	P	1.0	Single-precision real number
+54	S.2PID	I	10.0	Single-precision real number
+56	S.2PID	D	0.0	Single-precision real number
+58	S.2PID	GW	0.0	Single-precision real number
+60	S.2PID	GG	1.0	Single-precision real number
+62	S.OUT1	MVP	0.0	Single-precision real number
+64	S.2PID	$\alpha$	0.0	Single-precision real number
+66	S.2PID	$\beta$	1.0	Single-precision real number
⋮	⋮	⋮	⋮	⋮
+90	—	—	0.0	Single-precision real number
+92	—	—	0.0	Single-precision real number
+94	—	—	0.0	Single-precision real number






The signs in the ladder diagram mean as follows.



Item	Instruction name			
	S.IN	S.PHPL	S.2PID	S.OUT1
(1) Input data start device	R0	R20	R40	R60
(2) Block memory start device	R100	R120	R140	R160
(3) Operation constant start device	R200	Null character string ("")	R240	R260
(4) Loop tag memory start device	R1000			
(5) Set value start device	—	—	R300	—

## Data used by process control instructions



The data used by process control instructions includes the following.

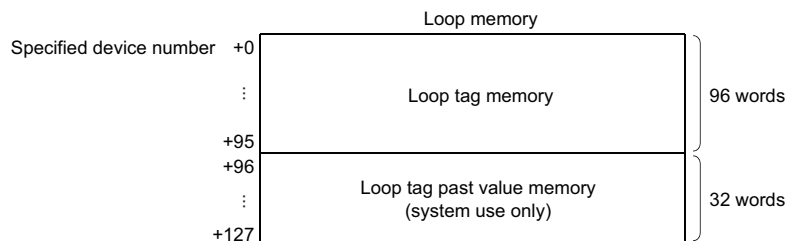
- Loop memory (  Page 565 Loop memory)
- Input data (  Page 566 Input data)
- Block memory (  Page 566 Block memory)
- Operation constant (  Page 567 Operation constant)
- Local work memory (  Page 567 Local work memory)

### Loop memory

The loop memory is an area in which the data used commonly by the process control instructions specified by the loop type is stored. The loop memory has also another area in which the data used by the CPU module system during execution of process control instructions is stored.

The loop memory consists of loop tag memory and loop tag past value memory.


-  Page 565 Loop tag memory
-  Page 565 Loop tag past value memory



**Point** 


The loop memory is configured with 128 words, and therefore a device which has 128 consecutive words should be specified when the loop memory area is allocated.

### ■ Loop tag memory

The loop tag memory is an area (96 words) in which the control information used commonly by the process control instructions specified by a loop type among the basic loop types (  Page 560 Basic loop types) is stored.

**Point** 

For the applications of the area used by process control instructions in the loop tag memory, refer to the following.

 Page 792 Lists of Loop Tag Memory

### ■ Loop tag past value memory

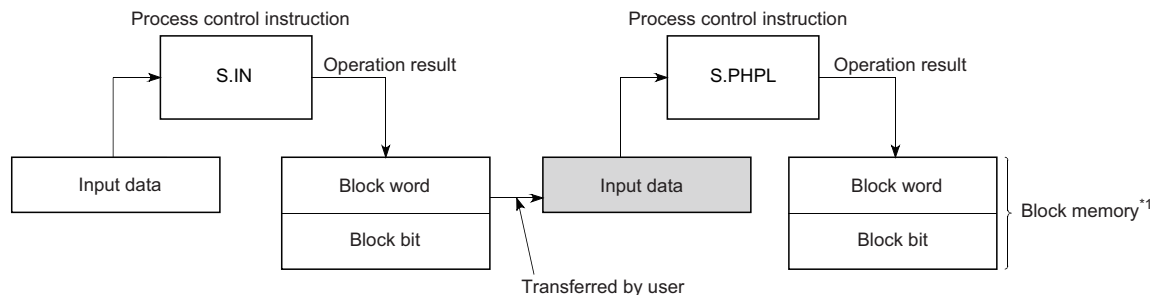
The loop tag past value memory is an area (32 words) used by the system during execution of process control instructions. No data can be written during operation. If data is written to the loop tag past value memory during operation, normal operation is disabled.

**Point** 

When starting a process control instruction, write 0 to the loop tag past value memory.

## Input data

Input data is variable data given to each process control instruction. The block word (BW) in the block memory in which the operation result of the previous process control instruction is stored is used as input data.



\*1 For the block memory, refer to the following.

☞ Page 566 Block memory

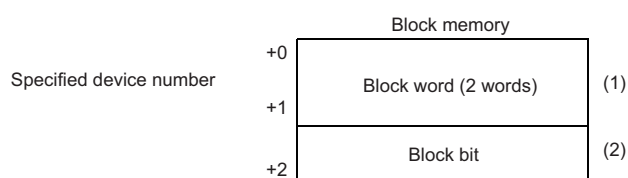
The application of input data varies depending on the instruction used. Refer to the descriptions of individual instructions.

## Block memory

The block memory is an area in which the output information of each process control instruction is stored.

The block memory consists of a block word (BW) and a block bit (BB).

- ☞ Page 566 Block word (BW)
- ☞ Page 566 Block bit (BB)

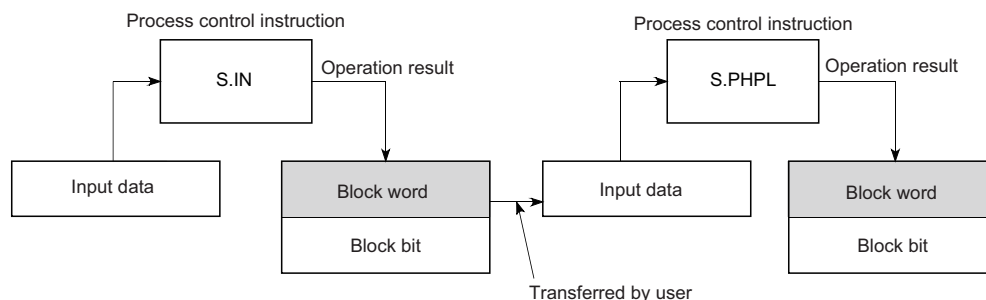


- (1) When storing a real number in the block word, use two words.
- (2) The block bit is used by setting each bit of one word to on or off.

The application of block memory varies depending on the instruction used. Refer to the descriptions of individual instructions.

### ■Block word (BW)

The block word (BW) is an area in which the operation result of each process control instruction is stored. The data stored in the block word (BW) is specified for the input data for the next process control instruction connected by the loop.



### ■Block bit (BB)

The block bit (BB) is an area in which alarm data during execution of each process control instruction is stored. Sixteen bits from b0 to b15 are represented as BB1 to BB16. If an alarm occurs in any of b1 to b15 (BB2 to BB16) used by instructions, 1 is stored in b0 (BB1). The bits that are not used are cleared to zero.

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BB16	BB15	BB14	BB13	BB12	BB11	BB10	BB9	BB8	BB7	BB6	BB5	BB4	BB3	BB2	BB1



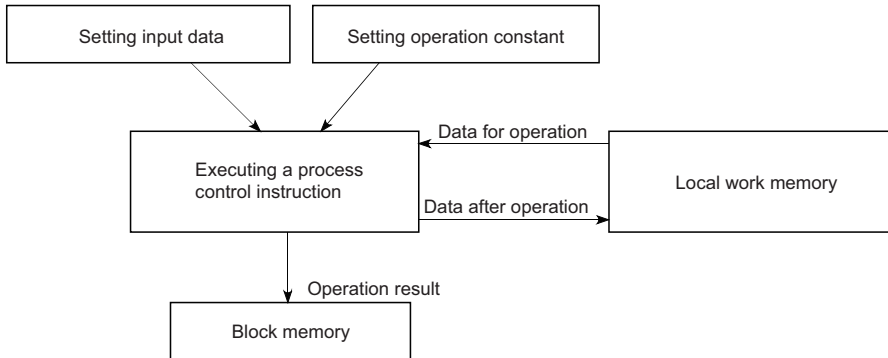
## Operation constant

The operation constant is an area in which the data used by only one process control instruction is stored.

The application of the operation constant varies depending on the instruction used. Refer to the descriptions of individual instructions.

## Local work memory

The local work memory is an area in which data is temporarily stored during operation of process control instructions.



The application and storage area of the local work memory vary depending on the instruction used. Refer to the descriptions of individual instructions.

# Loop tag memory assignments

Loop tag memory assignments are explained below.

**Ex.**


Loop tag memory assignments for two-degree-of-freedom PID control (S2PID)

Instruction used	Offset* <sup>1</sup>	Item	Recommended range	Standard value setting	Data type	Remarks
—	+0	—	—	—	—	Same values are used for all instructions in the same loop tag memory.
—	+1	MODE	0 to FFFFH	8H* <sup>2</sup>	16-bit unsigned binary	
—	+3	ALM	0 to FFFFH	4000H* <sup>2</sup>	16-bit unsigned binary	
—	+4	INH	0 to FFFFH	0H	16-bit unsigned binary	
S.PHPL	+10	PV	RL to RH	0.0* <sup>2</sup>	Single-precision real number	The offset values are fixed for each instruction.
S.OUT1	+12	MV	-10 to 110	0.0* <sup>2</sup>	Single-precision real number	
S.2PID	+14	SV	RL to RH	0.0	Single-precision real number	
S.2PID	+16	DV	-110 to 110	0.0* <sup>2</sup>	Single-precision real number	
S.OUT1	+18	MH	-10 to 110	100.0	Single-precision real number	
S.OUT1	+20	ML	-10 to 110	0.0	Single-precision real number	
S.PHPL	+22	RH	-999999 to 999999	100.0	Single-precision real number	
S.PHPL	+24	RL	-999999 to 999999	0.0	Single-precision real number	
S.PHPL	+26	PH	RL to RH	100.0	Single-precision real number	
S.PHPL	+28	PL	RL to RH	0.0	Single-precision real number	
S.PHPL	+30	HH	RL to RH	100.0	Single-precision real number	
S.PHPL	+32	LL	RL to RH	0.0	Single-precision real number	
S.IN	+38	$\alpha$	0 to 1	0.2	Single-precision real number	
S.PHPL	+40	HS	0 to 999999	0.0	Single-precision real number	
S.PHPL	+42	CTIM	0 to 999999	0.0	Single-precision real number	
S.PHPL	+44	DPL	0 to 999999	100.0	Single-precision real number	
S.2PID	+46	CT	0 to 100	1.0	Single-precision real number	
S.OUT1	+48	DML	0 to 100	100.0	Single-precision real number	
S.2PID	+50	DVL	0 to 100	100.0	Single-precision real number	
S.2PID	+52	P	0 to 999999	1.0	Single-precision real number	
S.2PID	+54	I	0 to 999999	10.0	Single-precision real number	
S.2PID	+56	D	0 to 999999	0.0	Single-precision real number	
S.2PID	+58	GW	0 to 100	0.0	Single-precision real number	
S.2PID	+60	GG	0 to 999999	1.0	Single-precision real number	
S.OUT1	+62	MVP	-999999 to 999999	0.0* <sup>2</sup>	Single-precision real number	
S.2PID	+64	$\alpha$	0 to 1	0.0	Single-precision real number	
S.2PID	+66	$\beta$	0 to 1	1.0	Single-precision real number	

\*1 This is the number of words from the start of the loop tag.

\*2 Values may change depending on the operation result.

For details on the loop tag memory assignments, refer to the following.

 Page 792 Lists of Loop Tag Memory

## Common items

This section describes the common items in loop types.

### ■ Alarm detection (ALM)

Alarm detection (ALM) indicates loop alarm information. The default setting is 4000H which indicates manual operation in loop stop state. To enable auto alarm, set it to 0000H.

b15	b14	...	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
	SPA			DMLA	OOPA	SEA	HHA	LLA	PHA	PLA	DPPA	DPNA	DVLA	MHA	MLA

Details of Alarm detection (ALM) are shown below.

Item	Name	Status	Set by
SPA	Stop alarm	Loop STOP state • Loop mode MAN is entered. • Loop stop processing is performed for the output value (BW) and alarm signal.	User
DMLA	Output variation rate limit alarm	As the result of checking input data using the variation rate limiter, the output variation rate limit has been exceeded.	System
OOPA	Output open alarm	The operation output signal is in open state due to disconnection.	System
SEA	Sensor alarm	Sensor error alarm	System
HHA	Upper upper limit alarm	The process value exceeds the upper limit defined for the process equipment.	System
LLA	Lower lower limit alarm	The process value underruns the lower limit defined for the process equipment.	System
PHA	Upper limit alarm	As the result of upper limit checking, the process value exceeds the upper limit.	System
PLA	Lower limit alarm	As the result of lower limit checking, the process value underruns the lower limit.	System
DPPA	Positive direction variation rate alarm	The variation rate exceeds the variation rate range on an upward trend.	System
DPNA	Negative direction variation rate alarm	The variation rate underruns the variation rate range on a downward trend.	System
DVLA	Large deviation alarm	The result of a deviation check shows that the deviation limit is exceeded. (The deviation check determines whether the deviation has been reduced completely underrunning the alarm value. To do so, when the deviation has been reduced to a certain value range from the alarm value, the large deviation alarm is released.)	System
MHA	Output upper limit alarm	As the result of checking with the upper/lower limiter, the value output by the limiter exceeds the output upper limit.	System
MLA	Output lower limit alarm	As the result of checking with the upper/lower limiter, the value output by the limiter underruns the output lower limit.	System

### ■ Disable alarm detection (INH)

This disable alarm detection of each item. The alarms whose detection is disabled by INH are not detected.

INH bits b0 to b11 correspond to ALM bits b0 to b11.

b15	...	b13	...	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ERRI		TRKF		DMLI	OOP	SEI	HHI	LLI	PHI	PLI	DPP	DPN	DVLI	MHI	MLI

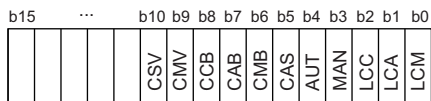
(2) (1)

(1) Tracking flag (Do not rewrite the tracking flag.)

(2) Alarm detection disabled

## ■Control mode (MODE)

The process control instructions have the following control modes that satisfy the following operations in a system connected to the operator station, programmable controller, host computer, and machine side operation panel. Set only 1 bit of flag for the control mode (MODE).



Details of the control mode (MODE) are given below.

Control mode	Description	Application
MAN (MANUAL)	<ul style="list-style-type: none"> <li>Manual operation from operator station</li> <li>SV and MV can be set.</li> </ul>	Used for monitoring and control from the operator station.
AUT (AUTOMATIC)	<ul style="list-style-type: none"> <li>Automatic operation</li> <li>SV can be set.</li> <li>MV cannot be set.</li> </ul>	
CAS (CASCADE)	<ul style="list-style-type: none"> <li>Cascade operation</li> <li>SV nor MV cannot be set.</li> </ul>	
CMV (COMPUTER MV)	Automatic MV setting from host computer	Loop control from the host computer is possible.
CSV (COMPUTER SV)	Automatic SV setting from host computer	Used to operate and monitor the control mode on the operator station.
CMB (COMPUTER MANUAL BACK UP)	Manual operation backup when the host computer is abnormal	When the computer fails during loop control by the host computer, backup is performed by the predetermined operator station.
CAB (COMPUTER AUTOMATIC BACK UP)	Automatic operation backup performed when the host computer is abnormal	
CCB (COMPUTER CASCADE BACK UP)	Cascade operation backup performed when the host computer is abnormal	
LCM (LOCAL MANIPULATED)	Local manual operation	
LCA (LOCAL AUTOMATIC)	Local automatic operation	When the plant is started, the operation and start-up are performed by using equipment such as the machine side operation panel away from the operator station, and the control mode is monitored at the operator station.
LCC (LOCAL CASCADE)	Local cascade operation	

# How to execute process control instructions

## Execution cycle and control cycle

### ■ Execution cycle

The execution cycle is an interval at which process control instructions are executed.

The following methods can be used to execute process control instructions in each execution cycle.

Execution method	Description
Timer	A timer is used to measure the execution cycle and a process control instruction is executed when the time of the timer is up.
Interrupt program	Interrupt programs I28 to I31 are each executed every execution cycle.
Fixed scan execution type program	Fixed scan execution type programs are each executed every execution cycle.

### Point

Store the execution cycle value set above in SD816 (execution cycle) and SD817 (execution cycle) as a single-precision real number, because it is used in process control instructions.

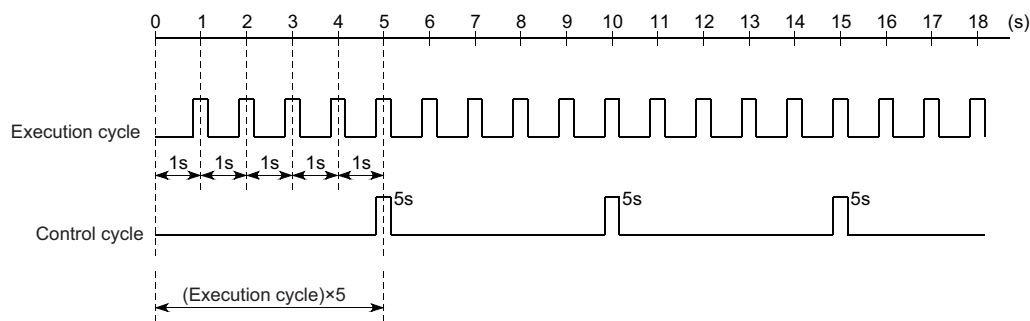
### ■ Control cycle

The control cycle is a cycle in which PID control is performed by instructions such as S.2PID. For the control cycle, specify an integral multiple of the execution cycle. Execution cycles are counted during execution, and PID operation is performed when the specified control cycle is reached.

Specify the control cycle used in the loop tag memory. Instructions such as S.2PID perform PID control based on the control cycle value specified in the loop tag memory.

### Ex.

The S.2PID instruction performs monitoring every second and implements PID control every 5 seconds.



### Point

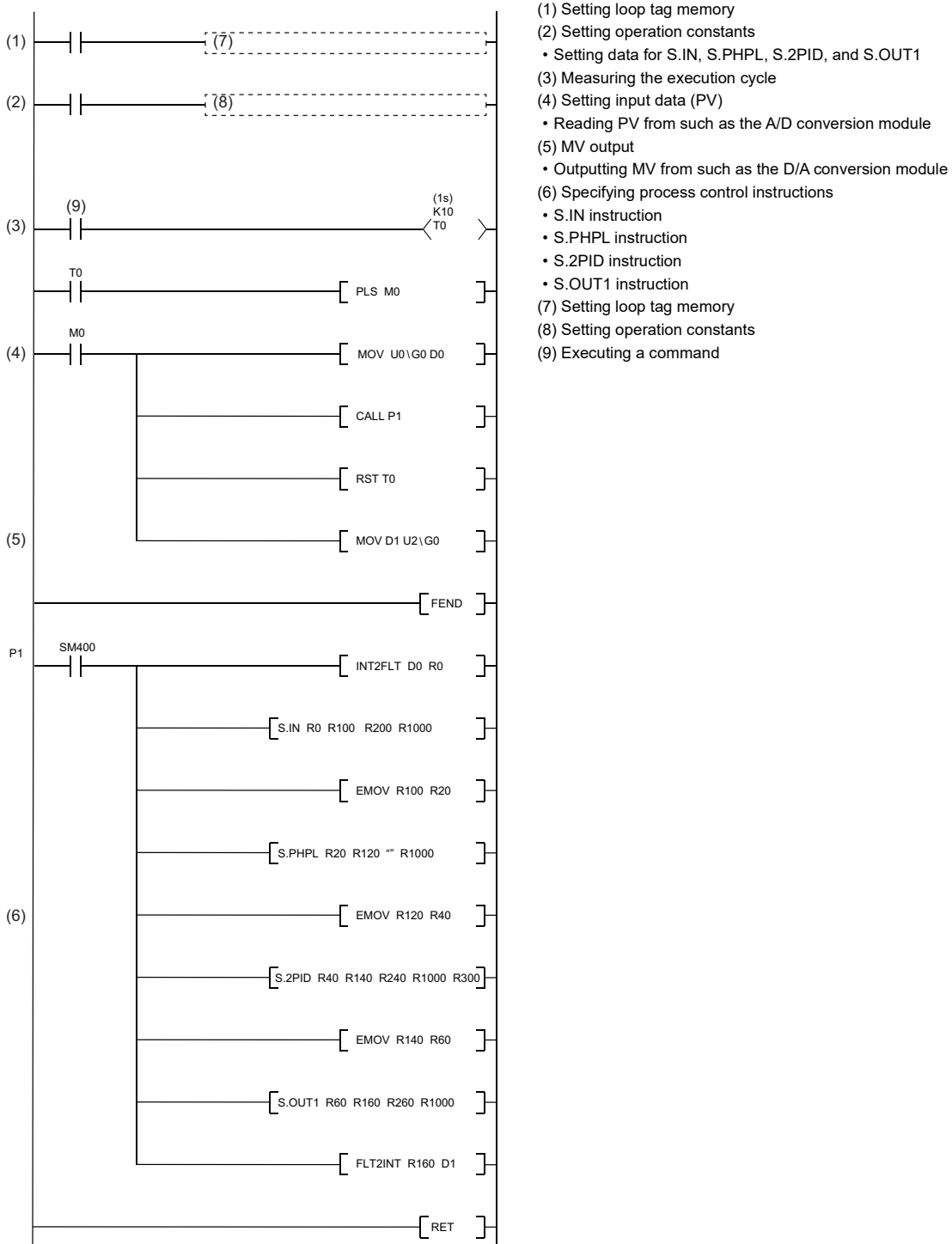
Setting the control cycle to an integral multiple of the execution cycle enables monitoring such as for checking the process value range every execution cycle.

## Concept of program

This section describes the concept of programs using process control instructions.

**Ex.**

Program example using the S.2PID instruction in an execution cycle of 1 second



For specific program examples using process control instructions, refer to the following.

☞ Page 807 Program Examples

# Execution condition switching

---

## Loop RUN/STOP

---

If any loop component such as a detector or operation terminal other than the programmable controller fails, each loop can be run and stopped independently for the purpose of maintenance. SPA of the alarm detection (ALM) is used to run/stop the applicable loop.

### ■Basic operation to stop a loop

- Output status is retained. (Example: Output of S.2PID instruction = 0)
- No alarm is detected.
- The control mode is MAN.

## Tracking

---

Tracking refers to making a certain signal follow and match another signal.

### Tracking function

---

The tracking function used by process control instructions includes the bumpless function and output limiter processing function.

#### ■Bumpless function

The bumpless function prevents manipulated value (MV) output stepping changes when switching from the automatic mode to manual mode, and continuously and smoothly controls MV output.

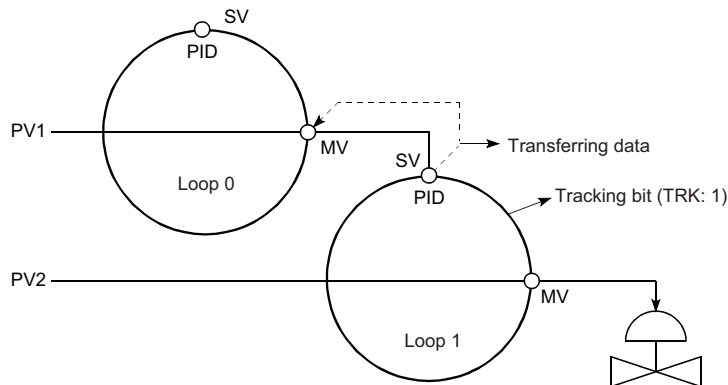
#### ■Output limiter processing function

The output limiter processing function limits the upper or lower limit of the manipulated value (MV) output by the PID operation in automatic mode. This output limiter processing function is only valid in automatic mode and is not executed in manual mode. The output limiter processing function is not executed either even in automatic mode if the tracking bit (TRK) is set to 0 (Disable tracking).

## Cascade loop tracking

The process control loops making up a cascade loop use the manipulated value (MV) of a primary loop (Loop 0) as the set value (SV) of a secondary loop (Loop 1). Tracking is performed to prevent a sudden change in the set value (SV) when the control mode of the secondary loop (Loop1) is changed.

Cascade loop tracking processing is outlined below.



In cascade operation, the manipulated value (MV) of Loop 0 is transferred to the set value (SV) of Loop 1. When cascade operation is not performed, the set value (SV) of Loop 1 is transferred to the manipulated value (MV) of Loop 0. (Tracking to the source specified as the input terminal of the set value (SV) of Loop 1)

Tracking is performed when the control mode is switched to one other than CAS, CSV, or CCB.

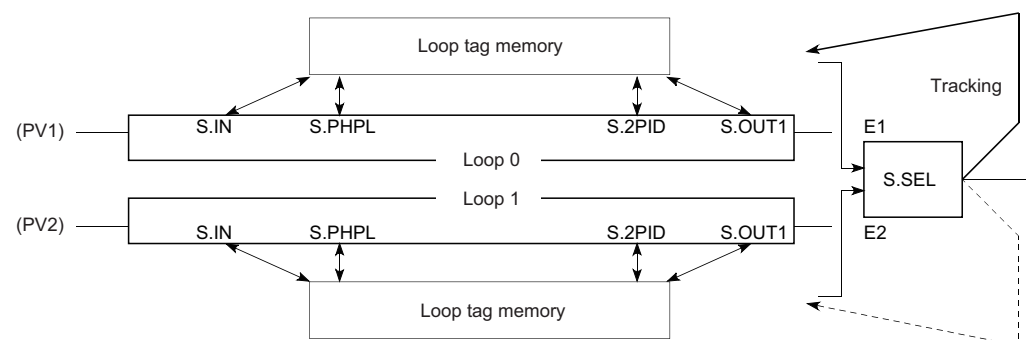
For S.2PID (two-degree-of-freedom PID control), set the following operation constant items to specify tracking.

Setting item		Setting
Tracking bit (TRK)		1 (Tracking performed)
Set value pattern (SVPTN)	Set value pattern	0 (Set value is upper loop MV.)
	Set value used	0 (E2 is specified.)

## Loop selector tracking

Tracking processing is performed under the following conditions.

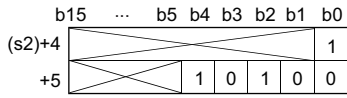
- The control mode is MAN, CMB, CMV, or LCM, and the tracking bit (TRK) is 1.
- The control mode is AUT, CAS, CAB, CCB, CSV, LCA, or LCC, and the tracking bit (TRK) is 1 and BB1 of the block bit (BB) is 1.





**Ex.**

When the input value (E1) of the S.SEL instruction uses the manipulated value (MV) of the upper loop (Loop 0), the manipulated value (MV) of the S.SEL instruction is tracked to the manipulated value (MV) of Loop 0. The S.SEL instruction specifies tracking according to the following operation constant items.



Operation constant	Bit position	Stored value	
(s2)+4	Tracking bit	b0	0: Tracking not performed 1: Tracking performed
(s2)+5	Set value pattern	b0	e1 0: E1 1: E2
		b1	E1 0: Use 1: Not use
		b2	E2 0: Use 1: Not use
		b3	E1 0: E1 is the upper loop MV. 1: E1 is not the upper loop MV.
		b4	E2 0: E2 is the upper loop MV. 1: E2 is not the upper loop MV.

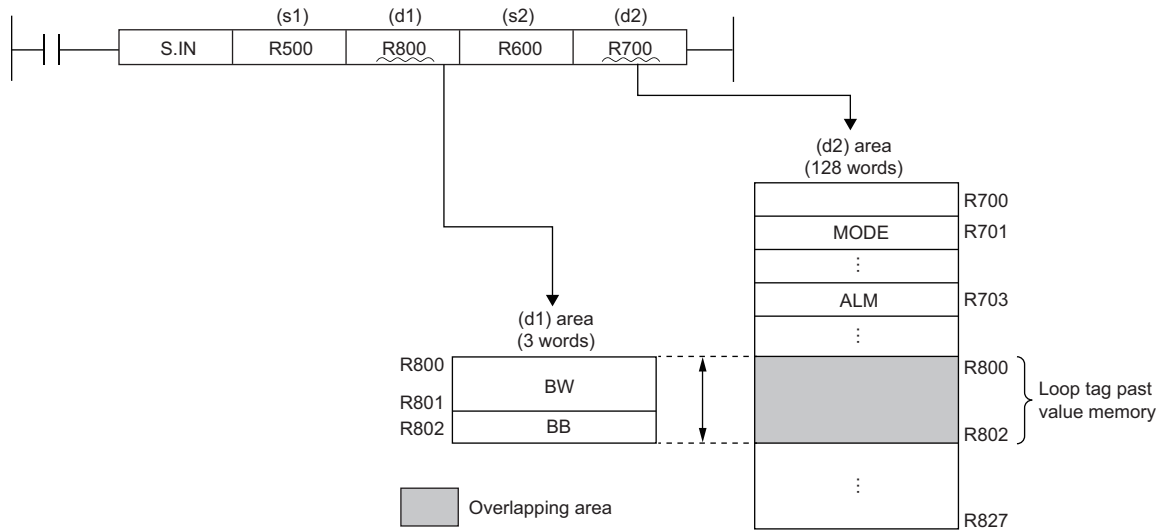
# Precautions

## Overlapping of specified data areas

The process control instructions check for area overlapping of input and output data. Specify input and output data after checking their areas are not overlapped.

**Ex.**

Example of area overlapping caused by a process control instruction



In the above example, the areas of (d1) and (d2) specified in the setting data are overlapping and an error results.

## Operand specification

Regardless of the program language used, specify devices for the operations of process control instructions. Do not specify labels.

## Errors of process control instructions

When an error occurs in process control instructions, the detailed information of the error is stored in SD81 to SD111 (detailed information 1) and SD113 to SD143 (detailed information 2). For the information to be stored, refer to the following.

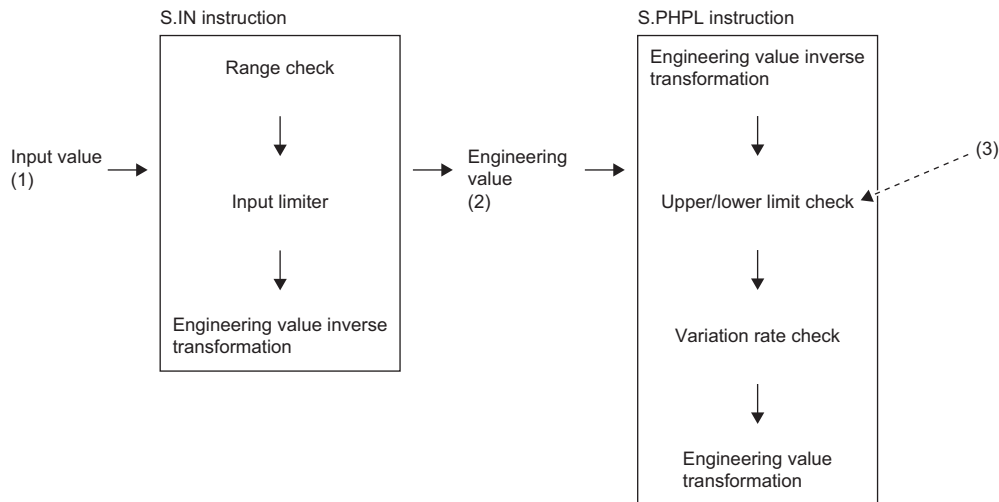
MELSEC iQ-R CPU Module User's Manual (Application)

## Arithmetic error of floating-point data

When using floating-point data, set error tolerances on floating-point operations. An arithmetic error may cause an unexpected alarm.

**Ex.**

An alarm due to an arithmetic error (in an execution of the S.PHPL instruction)



(1) A value exceeding the set value of the upper limit of the input limiter is input to the S.IN instruction.

(2) An output value (engineering value) of the S.IN instruction may not be exactly 100%. It can be slightly above 100%.

(3) When the upper limit value is set as 100% for the S.PHPL instruction, an alarm occurs. (Similarly, when a value less than the set value of the lower limit value of the input limiter is input to the S.IN instruction, an alarm can occur.)

To avoid an alarm occurrence due to an arithmetic error, set each value as the following example.

**Ex.**

Adding a program to disable alarm detection

Add a program that performs the following processing.

- When RH = HH, the bit HHI of the disable alarm detection (INH) turns on.
- When RH = PH, the bit PHI of the disable alarm detection (INH) turns on.
- When RL = LL, the bit LLI of the disable alarm detection (INH) turns on.
- When RL = PL, the bit PLI of the disable alarm detection (INH) turns on.

**Ex.**

Alarm value settings for the S.PHPL instruction

- 100.1% for the upper upper limit alarm value (HH)
- -0.1% for the lower lower limit alarm value (LL)
- 100.1% for the upper limit alarm value (PH)
- -0.1% for the lower limit alarm value (PL)

# 21.2 Lists of Process Control Instructions

## I/O control instructions

### ■Analog input processing

Instruction symbol	Processing details	Reference
S.IN	Performs following processing to the input data (PV): range check, input limiter, engineering value transformation, and digital filter.	Page 583 S.IN

### ■Output processing 1 with mode switching

Instruction symbol	Processing details	Reference
S.OUT1	Calculates MV (0 to 100%) from the input data ( $\Delta$ MV), and performs the variation rate & upper/lower limiter processing and output conversion processing.	Page 589 S.OUT1

### ■Output processing 2 with mode switching

Instruction symbol	Processing details	Reference
S.OUT2	Performs variation rate & upper/lower limiter processing and output conversion processing to the input data (MV).	Page 595 S.OUT2

### ■Manual output

Instruction symbol	Processing details	Reference
S.MOUT	Reads the manipulated value (MV) from the loop tag memory and performs output conversion.	Page 600 S.MOUT

### ■Time proportioning

Instruction symbol	Processing details	Reference
S.DUTY	Outputs ON and OFF by changing the ON/OFF ratio in a given cycle in proportion to the input data (0 to 100%).	Page 603 S.DUTY

### ■Batch counter

Instruction symbol	Processing details	Reference
S.BC	Compares the input data with the set value, and outputs bit data when it reaches the set value.	Page 609 S.BC

### ■Pulse integration

Instruction symbol	Processing details	Reference
S.PSUM	Integrates the number of input pulses, and outputs the operation result.	Page 613 S.PSUM

## Control operation instructions

### ■Basic PID control

Instruction symbol	Processing details	Reference
S.PID	Performs process value derivative type (inexact differential) PID operation. The instruction performs the following processing steps: SV setting, tracking, gain (Kp) operation, PID operation, and deviation check.	Page 618 S.PID

### ■Two-degree-of-freedom PID control

Instruction symbol	Processing details	Reference
S.2PID	Performs two-degree-of-freedom PID control operation (inexact differential). The instruction performs the following processing steps: SV setting, tracking, gain (Kp) operation, two-degree-of-freedom PID control operation, and deviation check.	Page 625 S.2PID

### ■Position type PID control

Instruction symbol	Processing details	Reference
S.PIDP	Performs position type PID operation. The instruction performs the following processing steps: SV setting, tracking, gain (Kp) operation, PID operation, deviation check, and control mode determination. Depending on the operation result up to the mode determination processing, the instruction performs either variation rate & upper/lower limiter and output conversion, or alarm clear and output conversion processing.	Page 632 S.PIDP

## ■Sample PI control

Instruction symbol	Processing details	Reference
S.SPI	Monitors the operating time (operating time (ST) or hold time (HT)), and performs the following processing steps: SV setting, tracking, gain (Kp) operation, SPI operation, and deviation check if the operating time is in ST.	Page 640 S.SPI

## ■I-PD control

Instruction symbol	Processing details	Reference
S.IPD	Performs I-PD operation. The instruction performs the following processing steps: SV setting, tracking, gain Kp operation, IPD operation, and deviation check.	Page 648 S.IPD

## ■Blend PI control

Instruction symbol	Processing details	Reference
S.BPI	Performs blend PI operation. The instruction performs the following processing steps: SV setting, tracking, gain Kp operation, BPI operation, and deviation check.	Page 655 S.BPI

## ■Ratio calculation

Instruction symbol	Processing details	Reference
S.R	Performs the following processing steps to the input data: engineering value transformation, tracking, variation rate limiter, and ratio calculation.	Page 662 S.R

## ■Upper/lower limit alarm

Instruction symbol	Processing details	Reference
S.PHPL	Performs the upper/lower limit check on the input data.	Page 667 S.PHPL

## ■Upper/lower limit alarm for power factor

Instruction symbol	Processing details	Reference
S.PHPL2	Performs the upper/lower limit check for power factor on the input data.	Page 673 S.PHPL2

## ■Lead-lag compensation

Instruction symbol	Processing details	Reference
S.LLAG	Performs lead-lag compensation to the input data, and outputs the operation result.	Page 681 S.LLAG

## ■Integral control

Instruction symbol	Processing details	Reference
S.I	Performs integration operation to the input data, and outputs the operation result.	Page 684 S.I

## ■Derivative control

Instruction symbol	Processing details	Reference
S.D	Performs differential operation to the input data, and outputs the operation result.	Page 686 S.D

## ■Dead time

Instruction symbol	Processing details	Reference
S.DED	Delays the output of the input data by the specified dead time.	Page 689 S.DED

## ■High selector

Instruction symbol	Processing details	Reference
S.HS	Outputs the maximum value of the input data.	Page 692 S.HS

## ■Low selector

Instruction symbol	Processing details	Reference
S.LS	Outputs the minimum value of the input data.	Page 694 S.LS

## ■Middle value selector

Instruction symbol	Processing details	Reference
S.MID	Outputs the intermediate value (the value between the maximum and minimum values) of the input data.	Page 696 S.MID

## ■Average value calculation

Instruction symbol	Processing details	Reference
S.AVE	Calculates the mean (average) value of the input data, and outputs the operation result.	Page 699 S.AVE

## ■Upper/lower limiter

Instruction symbol	Processing details	Reference
S.LIMIT	Applies a limiter with hysteresis to the output value.	Page 701 S.LIMIT

## ■Variation rate limiter 1

Instruction symbol	Processing details	Reference
S.VLMT1	Limits the variation speed when the variation rate of the input (E1) exceeds the limit.	Page 704 S.VLMT1

## ■Variation rate limiter 2

Instruction symbol	Processing details	Reference
S.VLMT2	Holds the last value when the variation rate of the input (E1) exceeds the limit.	Page 707 S.VLMT2

## ■Two-position (on/off) control

Instruction symbol	Processing details	Reference
S.ONF2	Performs two-position (on/off) control. The instruction performs the following processing steps: SV setting, tracking, MV correction, MV output, and two-position (on/off) control.	Page 710 S.ONF2

## ■Three-position (on/off) control

Instruction symbol	Processing details	Reference
S.ONF3	Performs three-position (on/off) control. The instruction performs the following processing steps: SV setting, tracking, MV correction, MV output, and three-position (on/off) control.	Page 716 S.ONF3

## ■Dead band

Instruction symbol	Processing details	Reference
S.DBND	Provides a dead band and performs output processing.	Page 722 S.DBND

## ■Program setter

Instruction symbol	Processing details	Reference
S.PGS	Provides control output according to the SV and MV patterns.	Page 724 S.PGS

## ■Loop selector

Instruction symbol	Processing details	Reference
S.SEL	Outputs the value selected by the selection signal from the input data in automatic mode, and outputs the manipulated value (MV) in the loop tag memory in manual mode.	Page 729 S.SEL

## ■Bumpless transfer

Instruction symbol	Processing details	Reference
S.BUMP	Gradually brings the output value (BW) closer to the output set value (E1) from the output control value (E2) when the mode switching signal (e1) changes from manual to automatic.	Page 735 S.BUMP

## ■Analog memory

Instruction symbol	Processing details	Reference
S.AMR	Increases or decreases the output value (BW) at a fixed rate.	Page 738 S.AMR

## Correction operation instructions

### ■Function generator

Instruction symbol	Processing details	Reference
S.FG	Outputs the input data values according to the specified function generator pattern.	Page 741 S.FG

### ■Inverse function generator

Instruction symbol	Processing details	Reference
S.IFG	Outputs the input data values according to the specified inverse function generator pattern.	Page 744 S.IFG

### ■Standard filter

Instruction symbol	Processing details	Reference
S.FLT	Outputs the mean (average) value of the 'n' pieces of data collected at the specified data collection intervals (ST).	Page 747 S.FLT

### ■Integration

Instruction symbol	Processing details	Reference
S.SUM	Integrates the input data, and outputs the operation result.	Page 750 S.SUM

### ■Temperature/pressure correction

Instruction symbol	Processing details	Reference
S.TPC	Performs temperature/pressure correction processing to the input data.	Page 752 S.TPC

### ■Engineering value transformation

Instruction symbol	Processing details	Reference
S.ENG	Performs engineering value transformation processing to the input data.	Page 755 S.ENG

### ■Engineering value inverse transformation

Instruction symbol	Processing details	Reference
S.IENG	Performs engineering value inverse transformation processing to the input data.	Page 757 S.IENG

## Arithmetic operation instructions

### ■Addition

Instruction symbol	Processing details	Reference
S.ADD	Adds input data with a coefficient.	Page 759 S.ADD

### ■Subtraction

Instruction symbol	Processing details	Reference
S.SUB	Subtracts input data with a coefficient.	Page 761 S.SUB

### ■Multiplication

Instruction symbol	Processing details	Reference
S.MUL	Multiplies input data with a coefficient.	Page 763 S.MUL

### ■Division

Instruction symbol	Processing details	Reference
S.DIV	Divides input data with a coefficient.	Page 765 S.DIV

### ■Square root

Instruction symbol	Processing details	Reference
S.SQR	Outputs the square root ( $\sqrt{\quad}$ ) of input data.	Page 767 S.SQR

### ■Absolute value

Instruction symbol	Processing details	Reference
S.ABS	Outputs the absolute value of input data.	Page 769 S.ABS

## Comparison operation instructions

### ■ Comparing data

Instruction symbol	Processing details	Reference
S.>	Compares input data, and outputs the comparison result.	Page 771 S.>
S.<		Page 773 S.<
S.=		Page 775 S.=
S.>=		Page 777 S.>=
S.<=		Page 779 S.<=

### Auto tuning

Instruction symbol	Processing details	Reference
S.AT1	Performs auto tuning to make initial setting of PID constants.	Page 784 S.AT1



# 21.3 I/O Control Instructions

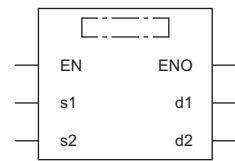
## Analog input processing

### S.IN

This instruction performs following processing to the input data (PV): range check, input limiter, engineering value transformation, and digital filter.

Ladder	ST
	ENO:=S_IN(EN,s1,s2,d1,d2);

### FBD/LD



### Execution condition

Instruction	Execution condition
S.IN	

### Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit


### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant				Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□\□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$		
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	
(d2)	—	—	○	—	—	—	—	○	—	—	—	—	

### Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value	-999999 to 999999	Single-precision real number	—	User
+1						

## ■Block memory

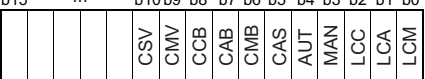
Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999 [%]	Single-precision real number	—	System
+2	BB	BB1 Alarm	b15                      ...                      b2 b1 b0  0: No alarm 1: Alarm	16-bit unsigned binary	—	System
		BB2 Input upper limit alarm				
		BB3 Input lower limit alarm				

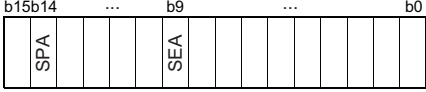
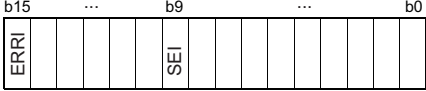
## ■Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	EMAX	Engineering value transformation upper limit	-999999 to 999999 [%]	Single-precision real number	100.0	User
+2 +3	EMIN	Engineering value transformation lower limit	-999999 to 999999 [%]	Single-precision real number	0.0	User
+4 +5	NMAX	Input upper limit	-999999 to 999999	Single-precision real number	100.0	User
+6 +7	NMIN	Input lower limit	-999999 to 999999	Single-precision real number	0.0	User
+8 +9	HH	Upper limit range error occurrence	-999999 to 999999	Single-precision real number	110.0	User
+10 +11	H	Upper limit range error return	-999999 to 999999	Single-precision real number	100.0	User
+12 +13	L	Lower limit range error return	-999999 to 999999	Single-precision real number	0.0	User
+14 +15	LL	Lower limit range error occurrence	-999999 to 999999	Single-precision real number	-10.0	User

## ■Loop tag memory

The loop tag memory occupies 96 words from the specified start device.

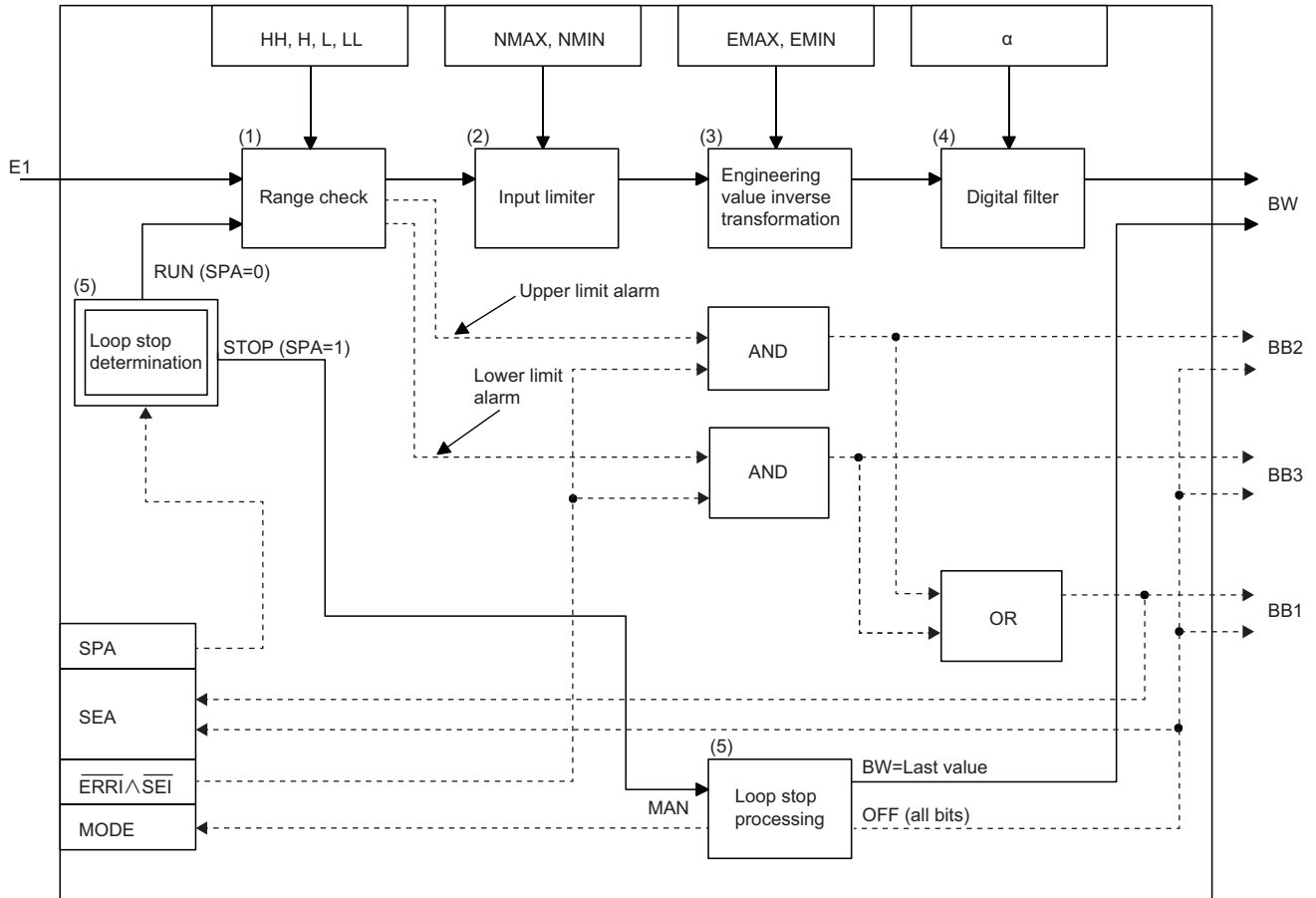
Operand: (d2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+1	MODE	Control mode	0 to FFFFH b15                      ...                      b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 	16-bit unsigned binary	8H	User/system

Operand: (d2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+3	ALM	Alarm detection	0 to FFFFH  SPA 0: Loop RUN 1: Loop STOP SEA 0: No alarm 1: Alarm	16-bit unsigned binary	4000H	User/ system
+4	INH	Disable alarm detection	0 to FFFFH  0: Alarm detection enabled 1: Alarm detection disabled	16-bit unsigned binary	0H	User/ system
+38 +39	$\alpha$	Filter coefficient	0 to 1	Single-precision real number	0.2	User

## Processing details

This instruction performs engineering value transformation of the input value (E1) in the device specified by (s1), and stores the result in the device specified by (d1). The instruction also performs input value (E1) range check, input limiter, and digital filter processing.

The following is the processing block diagram of the S.IN instruction. (The numbers (1) to (5) in the diagram indicate the order of the processing.)



### Range check (1)

- The instruction checks the range of the input value (E1).

If the input value (E1) goes beyond the upper or lower limits, an alarm is output.

Range check	Condition	Range check result (alarm output)		
		BB2	BB3	BB1, SEA
Upper limit check	$E1 \geq HH$	1*1	—	1*1
	$E1 \leq H$	0	—	0
	$H < E1 < HH$	Last value	—	Last value
Lower limit check	$E1 \leq LL$	—	1*1	1*1
	$E1 \geq L$	—	0	0
	$LL < E1 < L$	—	Last value	Last value

\*1 When ERR1 or SE1 of Disable alarm detection (INH) is set to 1, alarm output is disabled and therefore SEA of Alarm detection (ALM), Alarm (BB1), Input upper limit alarm (BB2), and Input lower limit alarm (BB3) are set to 0.

- Last value hold processing

When a range excess occurs (BB1 = 1) in the range check, whether to continue operation or terminate the S.IN instruction is determined by whether SM816 is on or off.

Condition	Description
SM816 is off (not in hold mode)	"Input limiter (2)" is performed even if a range excess occurs (BB1 = 1).
SM816 is on (in hold mode)	If a range excess occurs (BB1 = 1), the following processing is performed to terminate the S.IN instruction. <ul style="list-style-type: none"> <li>The last output value (BW) is held.</li> <li>Error information is stored in BB.</li> </ul>

### Input limiter (2)

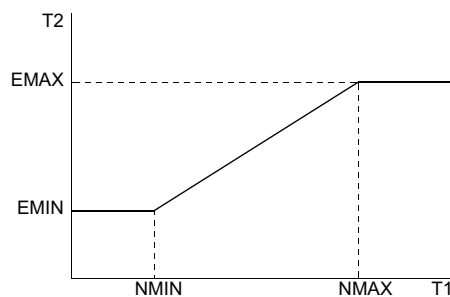
The instruction sets the upper and lower limits for the input value (E1).

Condition	Result (T1)
$E1 \geq NMAX$	NMAX
$E1 \leq NMIN$	NMIN
$NMIN < E1 < NMAX$	E1

### Engineering value inverse transformation (3)

The instruction performs engineering value inverse transformation of the input limiter result (T1) according to the following expression.

$$T2 = (EMAX - EMIN) \times \frac{T1 - NMIN}{NMAX - NMIN} + EMIN$$



### Digital filter (4)

The instruction applies a digital filter to the input value (E1) according to the following expression. The digital filter is used to reduce the effect of noise.

$$BW = T2 + \alpha \times (\text{last BW value} - T2)$$

## ■ Loop stop processing (5)

The following processing is performed according to the SPA status of the alarm detection (ALM).

SPA status	Processing details
1	Performs the following operations and terminates the S.IN instruction. <ul style="list-style-type: none"><li>• The last output value (BW) is held.</li><li>• The control mode (MODE) is set to MAN.</li><li>• All of the alarm bits (BB1, BB2, and BB3) are set to 0.</li><li>• ALM SEA is set to 0.</li></ul>
0	Performs "range check (1)".

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1), (d1), (s2), or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Upper limit range error occurrence (HH) < upper limit range error return (H), lower limit range error return (L) < lower limit range error occurrence (LL), or input upper limit (NMAX) < input lower limit (NMIN)

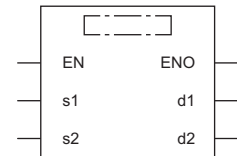
# Output processing 1 with mode switching

## S.OUT1

This instruction calculates MV (0 to 100%) from input data ( $\Delta MV$ ), performs variation rate & upper/lower limiter processing, and output conversion.

Ladder	ST
	<pre>ENO:=S_OUT1(EN,s1,s2,d1,d2);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.OUT1	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

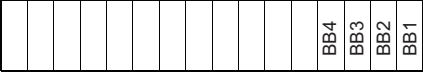
#### Applicable devices

Operand	Bit		Word		Z	Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□(H)G□		LT, LST, LC	LZ		K, H, E, \$			
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—

#### Input data

Operand: (s1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value ( $\Delta MV$ )	-999999 to 999999 [%]	Single-precision real number	—	User
+1						

## ■Block memory

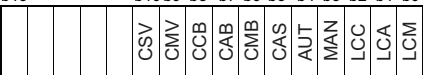
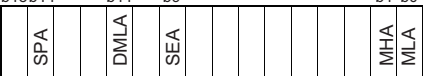
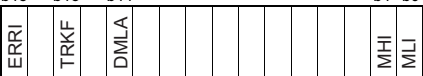
Operand: (d1)														
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by								
+0 +1	BW	Output value	-999999 to 999999	Single-precision real number	—	System								
+2	BB	<table border="1"> <tr> <td>BB1</td> <td>Alarm</td> </tr> <tr> <td>BB2</td> <td>Output upper limit alarm</td> </tr> <tr> <td>BB3</td> <td>Output lower limit alarm</td> </tr> <tr> <td>BB4</td> <td>Output variation rate alarm</td> </tr> </table>	BB1	Alarm	BB2	Output upper limit alarm	BB3	Output lower limit alarm	BB4	Output variation rate alarm	b15 ... b3 b2 b1 b0  0: No alarm 1: Alarm	16-bit unsigned binary	—	System
BB1	Alarm													
BB2	Output upper limit alarm													
BB3	Output lower limit alarm													
BB4	Output variation rate alarm													

## ■Operation constant

Operand: (s2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	NMAX	Output conversion upper limit	-999999 to 999999	Single-precision real number	100.0	User
+2 +3	NMIN	Output conversion lower limit	-999999 to 999999	Single-precision real number	0.0	User

## ■Loop tag memory

The loop tag memory occupies 96 words from the specified start device.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+1	MODE	Control mode	0 to FFFFH b15 ... b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 	16-bit unsigned binary	8H	User/ system
+3	ALM	Alarm detection	0 to FFFFH b15b14 ... b11 ... b9 ... b1 b0  SPA 0: Loop RUN 1: Loop STOP DMLA, SEA, MHA, MLA 0: No alarm 1: Alarm	16-bit unsigned binary	4000H	User/ system
+4	INH	Disable alarm detection	0 to FFFFH b15 ... b13 ... b11 ... b1 b0  TRKF 0: Tracking not set 1: Tracking set ERRI, DMLI, MHI, MLI 0: Alarm detection enabled 1: Alarm detection disabled	16-bit unsigned binary	0H	User/ system
+12 +13	MV	Manipulated value	-10 to 110 [%]	Single-precision real number	0.0	User/ system
+18 +19	MH	Output upper limit value	-10 to 110 [%]	Single-precision real number	100.0	User




Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+20 +21	ML	Output lower limit value	-10 to 110 [%]	Single-precision real number	0.0	User
+48 +49	DML	Output variation rate limit value	0 to 100 [%]	Single-precision real number	100.0	User
+54 +55	I	Integral constant	0 to 999999 [s]	Single-precision real number	10.0	User
+62 +63	MVP	MV internal operation value	-999999 to 999999 [%]	Single-precision real number	0.0	System

**Loop tag past value memory**

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

The loop tag past value memory occupies 32 words after the loop tag memory.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
(d2)+116	ALM2	Alarm detection 2	b15 ... b3 b2 b1 b0  MHA2, MLA2, DMLA2 0: No alarm 1: Alarm	—	—	System

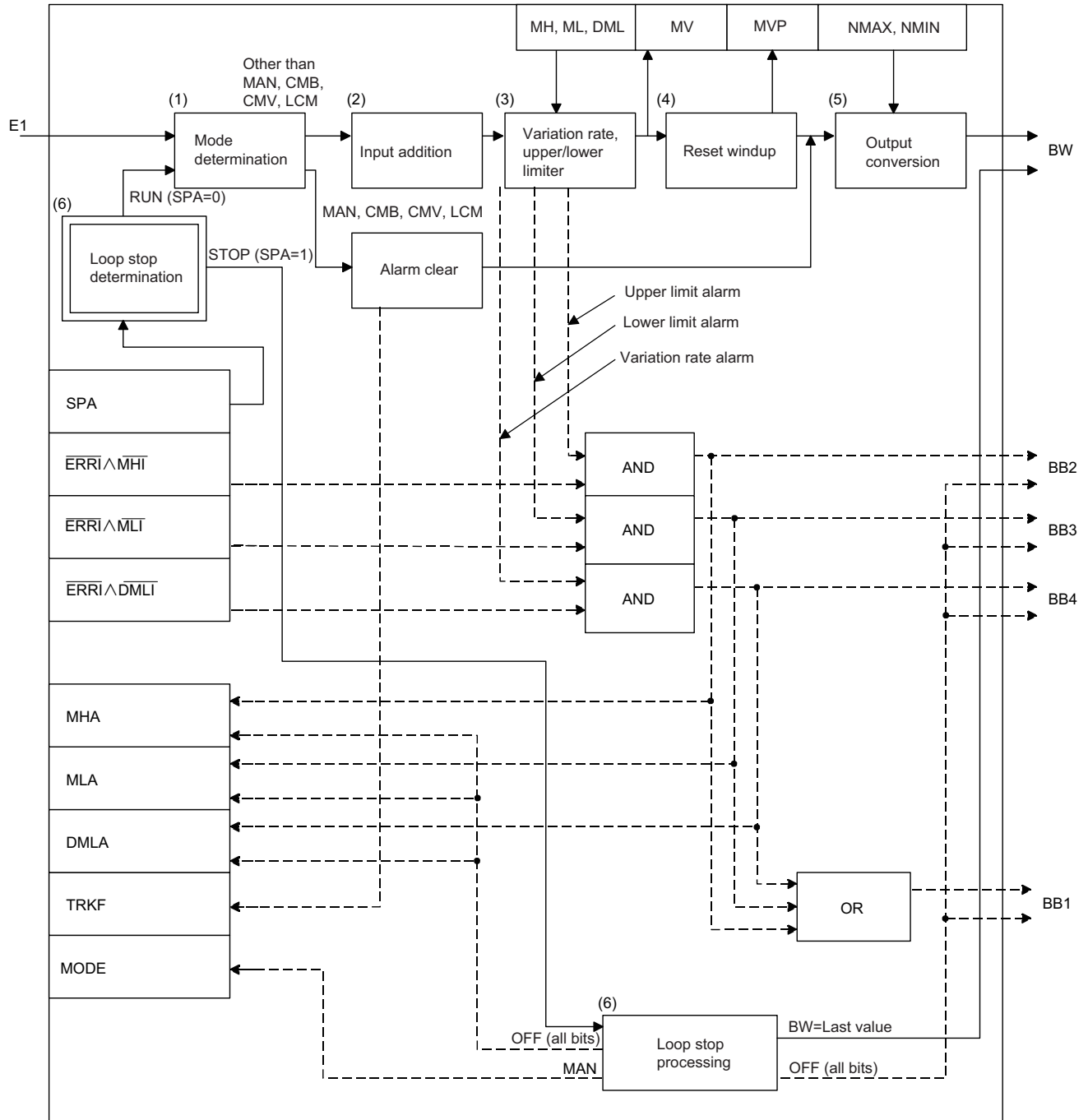
**Execution cycle (ΔT)**

Set the execution cycle (ΔT) in SD816 and SD817 in single-precision real number.

## Processing details

This instruction calculates the manipulated value (MV) from the input value ( $E1=\Delta MV$ ) in the device specified by (s1), and stores the result in the device specified by (d1). The instruction also performs variation rate & upper/lower limiter, reset windup, and output conversion processing of the calculated manipulated value (MV).

The following is the processing block diagram of the S.OUT1 instruction. (The numbers (1) to (6) in the diagram indicate the order of the processing.)



## ■ Mode determination (1)

The following processing is performed depending on the control mode (MODE).

Control mode (MODE)	Processing details
MAN, CMB, CMV, LCM (alarm clear processing)	<ul style="list-style-type: none"> <li>The MHA, MLA, and DMLA of the alarm detection (ALM) are set to 0.</li> <li>The MHA2 and MLA2 of the alarm detection 2 (ALM2) are set to 0.</li> <li>The alarm bits (BB1, BB2, BB3, and BB4) are set to 0.</li> <li>TRKF of INH is set to 1.</li> <li>"Output conversion processing (5)" is performed and the instruction ends.</li> </ul>
AUT, CAB, CAS, CCB, CSV, LCA, LCC	"Input addition processing (2)" is performed. However, when ALM SEA is 1 and SM817 is on, alarm bits BB1, BB2, BB3, and BB4 are set to 0 and the S.OUT1 instruction is terminated.

## ■ Input addition processing (2)

A temporary MV(T) is calculated on the basis of the input value ( $E1=\Delta MV$ ). The following processing is performed depending on the TRKF of INH.

Tracking flag (TRKF)	Processing details
1	<ul style="list-style-type: none"> <li>The manipulated value (MV) is stored in the MV internal operation value (MVP).</li> <li>The input value (E1) is set to 0. (<math>\Delta MV=0</math>)</li> <li>TRKF of INH is set to 0.</li> <li>A temporary MV(T) is calculated according to the following expression.  <math>T=E1+MVP</math>  <math>MVP=T</math> </li> </ul>
0	A temporary MV(T) is calculated according to the following expression. $T=E1+MVP$ $MVP=T$

## ■ Variation rate & upper/lower limiter (3)

The variation rate and upper/lower limits of the input value ( $E1=\Delta MV$ ) are checked, and the data after the processing and an alarm are output.

- Variation rate limiter processing performs the following operations, and outputs the result to the output variation rate alarm (BB4) and the DMLA of the alarm detection (ALM).

Condition	BB4, DMLA	Result (T1)
$ T-MV  \leq DML$	0	T
$(T - MV) > DML$	$1^{*1}$	MV + DML
$(T - MV) < -DML$	$1^{*1}$	MV - DML

\*1 If the DMLI or ERRI of the disable alarm detection (INH) is set to 1, the output variation rate alarm (BB4) and the DMLA of the alarm detection (ALM) are set to 0.

- Upper/lower limiter processing performs the following operations, and outputs the result to the output upper limit alarm (BB2); output lower limit alarm (BB3); MHA and MLA of the alarm detection (ALM); and MHA2 and MLA2 of the alarm detection (ALM2).

Condition	BB3, MLA, MLA2	BB2, MHA, MHA2	MV
$T1 > MH$	0	$1^{*2}$	MH
$T1 < ML$	$1^{*3}$	0	ML
$ML \leq T1 \leq MH$	0	0	T1

\*2 If the MHI or ERRI of the disable alarm detection (INH) is set to 1, the output upper limit alarm (BB2) and the MHA of the alarm detection (ALM) are set to 0.

Note that the MHA2 of the alarm detection 2 (ALM2) remains 1.

\*3 If the MLI or ERRI of the disable alarm detection (INH) is set to 1, the output lower limit alarm (BB3) and the MLA of the alarm detection (ALM) are set to 0.

Note that the MHA2 of the alarm detection 2 (ALM2) remains 1.

## ■Reset windup (4)

If the manipulated value (MV) goes beyond the upper or lower limit, the following operation is performed to return it to the upper or lower limit and enable immediate response when the deviation is inverted. However, when the integral constant (I) is 0, reset windup processing is not performed.

Condition	Operational expression
$T1 > MH, \frac{\Delta T}{I} \leq 1$	$MVP = (\frac{\Delta T}{I})(MH - T) + T$
$T1 < ML, \frac{\Delta T}{I} \leq 1$	$MVP = (\frac{\Delta T}{I})(ML - T) + T$

## ■Output conversion processing (5)

The output value (BW) is calculated from the following expression.

$$BW = \frac{NMAX - NMIN}{100} \times MV + NMIN$$

## ■Loop stop processing (6)

The following processing is performed according to the SPA status of the alarm detection (ALM).

SPA status	Processing details
1	The loop stops. When the loop stops, the following operations are performed and the S.OUT1 instruction ends. <ul style="list-style-type: none"> <li>• The last output value (BW) is held.</li> <li>• The DMLA, MHA, and DLA of the alarm detection (ALM) are set to 0.</li> <li>• The MHA2 and MLA2 of the alarm detection 2 (ALM2) are set to 0.</li> <li>• The control mode (MODE) is set to MAN.</li> <li>• The alarm bits (BB1, BB2, BB3, and BB4) are set to 0.</li> </ul>
0	The loop runs and "mode determination (1)" is performed.

## ■Hold processing (7)

This processing specifies whether to hold the output value (BW) by the S.OUT1 instruction when a sensor error occurs (detected by the S.IN instruction). The hold processing is performed when the value is determined as RUN by "Loop Stop Determination". SM817 is used to specify whether to hold the manipulated value (MV) when a sensor alarm occurs.

- SM817 = OFF: Do not hold the manipulated value (MV).
- SM817 = ON: Hold the manipulated value (MV).

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1), (s2), or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# Output processing 2 with mode switching

## S.OUT2

This instruction performs variation rate & upper/lower limiter processing and output conversion on the basis of input data (MV).

Ladder	ST
	ENO:=S_OUT2(EN,s1,s2,d1,d2);

### FBD/LD



### Execution condition

Instruction	Execution condition
S.OUT2	

### Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

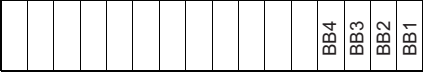
### Applicable devices

Operand	Bit		Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	○	—	—	—	—	
(d2)	—	—	○	—	—	—	○	—	—	—	—	

### Input data

Operand: (s1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value (MV)	-999999 to 999999 [%]	Single-precision real number	—	User
+1						

## ■Block memory

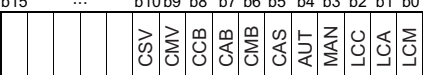
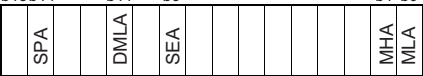
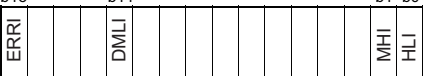
Operand: (d1)														
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by								
+0 +1	BW	Output value	-999999 to 999999	Single-precision real number	—	System								
+2	BB	<table border="1"> <tr> <td>BB1</td> <td>Alarm</td> </tr> <tr> <td>BB2</td> <td>Output upper limit alarm</td> </tr> <tr> <td>BB3</td> <td>Output lower limit alarm</td> </tr> <tr> <td>BB4</td> <td>Output variation rate alarm</td> </tr> </table>	BB1	Alarm	BB2	Output upper limit alarm	BB3	Output lower limit alarm	BB4	Output variation rate alarm	b15                      ...                      b3   b2   b1   b0  0: No alarm 1: Alarm	16-bit unsigned binary	—	System
BB1	Alarm													
BB2	Output upper limit alarm													
BB3	Output lower limit alarm													
BB4	Output variation rate alarm													

## ■Operation constant

Operand: (s2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	NMAX	Output conversion upper limit	-999999 to 999999	Single-precision real number	100.0	User
+2 +3	NMIN	Output conversion lower limit	-999999 to 999999	Single-precision real number	0.0	User

## ■Loop tag memory

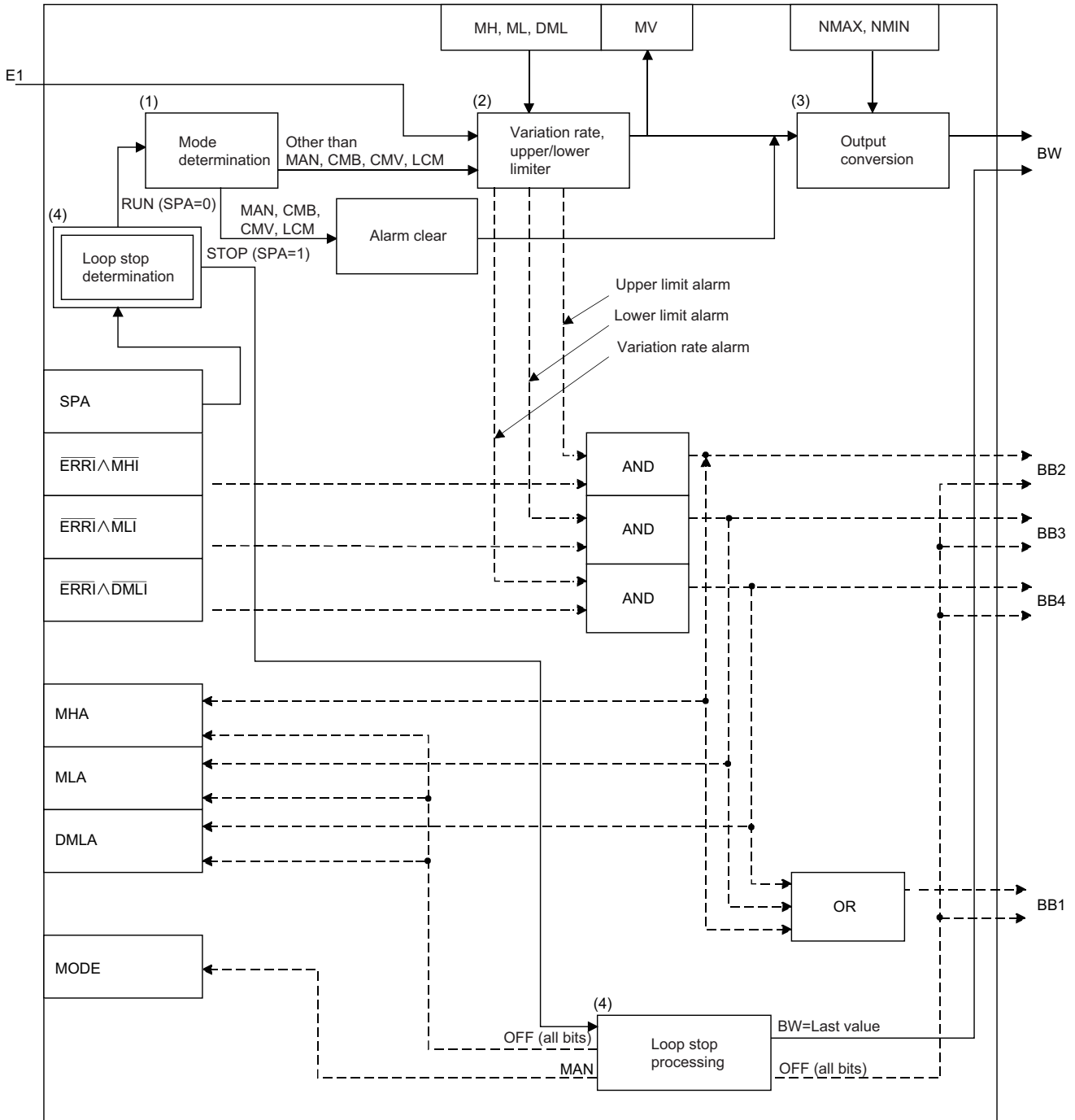
The loop tag memory occupies 96 words from the specified start device.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+1	MODE	Control mode	0 to FFFFH b15                      ...                      b10 b9   b8   b7   b6   b5   b4   b3   b2   b1   b0 	16-bit unsigned binary	8H	User/ system
+3	ALM	Alarm detection	0 to FFFFH b15b14                      ...                      b11                      ...                      b1                      b0  SPA 0: Loop RUN 1: Loop STOP DMLA, SEA, MHA, MLA 0: No alarm 1: Alarm	16-bit unsigned binary	4000H	User/ system
+4	INH	Disable alarm detection	0 to FFFFH b15                      ...                      b11                      ...                      b1                      b0  0: Alarm detection enabled 1: Alarm detection disabled	16-bit unsigned binary	0H	User/ system
+12 +13	MV	Manipulated value	-10 to 110 [%]	Single-precision real number	0.0	User/ system
+18 +19	MH	Output upper limit value	-10 to 110 [%]	Single-precision real number	100.0	User
+20 +21	ML	Output lower limit value	-10 to 110 [%]	Single-precision real number	0.0	User
+48 +49	DML	Output variation rate limit value	0 to 100 [%]	Single-precision real number	100.0	User

## Processing details

This instruction performs output conversion of the input value (E1=MV) in the device specified by (s1), and stores the result in the device specified by (d1). The instruction also performs variation rate & upper/lower limiter and output conversion processing of the input value at that time.

The following is the processing block diagram of the S.OUT2 instruction. (The numbers (1) to (4) in the diagram indicate the order of the processing.)



## ■ Mode determination (1)

The following processing is performed depending on the control mode (MODE).

Control mode (MODE)	Processing details
MAN, CMB, CMV, LCM (alarm clear processing)	<ul style="list-style-type: none"> <li>The MHA, MLA, and DMLA of the alarm detection (ALM) are set to 0.</li> <li>The alarm bits (BB1, BB2, BB3, and BB4) are set to 0.</li> <li>"Output conversion processing (3)" is performed and the instruction ends.</li> </ul>
AUT, CAB, CAS, CCB, CSV, LCA, LCC	"Variation rate & upper/lower limiter processing (2)" is performed. However, when ALM SEA is 1 and SM817 is on, alarm bits BB1, BB2, BB3, and BB4 are set to 0 and the S.OUT2 instruction is terminated.

## ■ Variation rate & upper/lower limiter (2)

The variation rate and upper/lower limits of the input value ( $E1=\Delta MV$ ) are checked, and the data after the processing and an alarm are output.

- Variation rate limiter processing performs the following operations, and outputs the result to the output variation rate alarm (BB4) and the DMLA of the alarm detection (ALM).

Condition	BB4, DMLA	Result (T1)
$ E1-MV  \leq DML$	0	E1
$(E1-MV) > DML$	1*1	MV + DML
$(E1-MV) < -DML$	1*1	MV - DML

\*1 If the DMLI or ERRI of the disable alarm detection (INH) is set to 1, the output variation rate alarm (BB4) and the DMLA of the alarm detection (ALM) are set to 0.

- Upper/lower limiter processing performs the following operations, and outputs the result to the output upper limit alarm (BB2); output lower limit alarm (BB3); and MHA and MLA of the alarm detection (ALM).

Condition	BB3, MLA	BB2, MHA	MV
$T1 > MH$	0	1*2	MH
$T1 < ML$	1*3	0	ML
$ML \leq T1 \leq MH$	0	0	T1

\*2 If the MHI or ERRI of the disable alarm detection (INH) is set to 1, the output upper limit alarm (BB2) and the MHA of the alarm detection (ALM) are set to 0.

\*3 If the MLI or ERRI of the disable alarm detection (INH) is set to 1, the output lower limit alarm (BB3) and the MLA of the alarm detection (ALM) are set to 0.

## ■ Output conversion processing (3)

The output value (BW) is calculated from the following expression.

$$BW = \frac{NMAX - NMIN}{100} \times MV + NMIN$$

## ■ Loop stop processing (4)

The following processing is performed according to the SPA status of the alarm detection (ALM).

SPA status	Processing details
1	The loop stops. When the loop stops, the following operations are performed and the S.OUT2 instruction ends. <ul style="list-style-type: none"> <li>The last output value (BW) is held.</li> <li>The DMLA, MHA, and DLA of the alarm detection (ALM) are set to 0.</li> <li>The control mode (MODE) is set to MAN.</li> <li>The alarm bits (BB1, BB2, BB3, and BB4) are set to 0.</li> </ul>
0	The loop runs and "mode determination (1)" is performed.

## ■ Hold processing (5)

This processing specifies whether to hold the output value (BW) by the S.OUT2 instruction when a sensor error occurs (detected by the S.IN instruction). The hold processing is performed when the value is determined as RUN by "Loop Stop Determination". SM817 is used to specify whether to hold the manipulated value (MV) when a sensor alarm occurs.

- SM817 = OFF: Do not hold the manipulated value (MV).
- SM817 = ON: Hold the manipulated value (MV).



## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1), (s2), or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

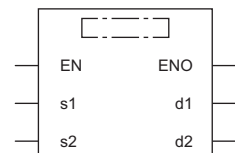
# Manual output

## S.MOUT

This instruction reads the manipulated value (MV) from the loop tag memory and performs output conversion.

Ladder	ST
	<pre>ENO:=S_MOUT2(EN,s1,s2,d1,d2);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.MOUT	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
(d1)	Block memory start device	Refer to "Block memory".	Single-precision real number
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

#### Applicable devices

Operand	Bit		Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□□G□, J□□□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	—	—	—	—	—	—	—	—	○	—
(d1)	—	—	○	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—	—
(d2)	—	—	○	—	—	—	○	—	—	—	—	—

#### Block memory

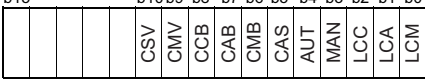

Operand: (d1)							
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by	
+0	BW	Output value	-999999 to 999999	Single-precision real number	—	System	
+1							

## ■ Operation constant

Operand: (s2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	NMAX	Output conversion upper limit	-999999 to 999999	Single-precision real number	100.0	User
+2 +3	NMIN	Output conversion lower limit	-999999 to 999999	Single-precision real number	0.0	User

## ■ Loop tag memory

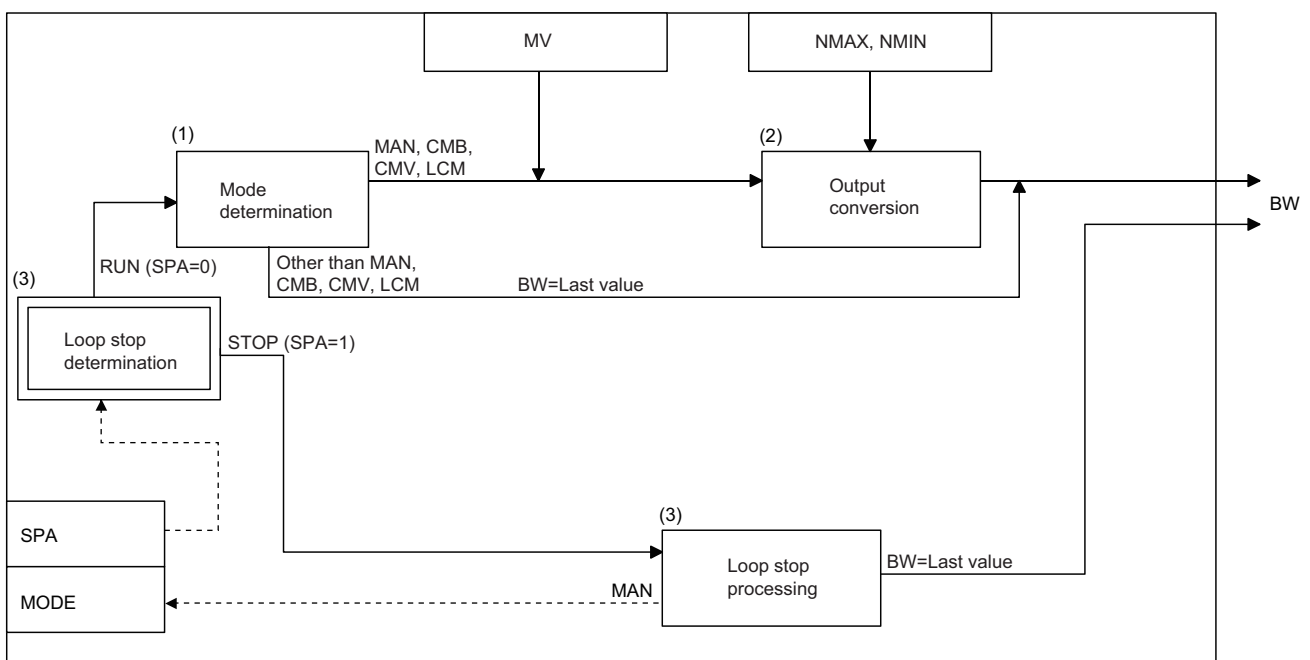
The loop tag memory occupies 96 words from the specified start device.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+1	MODE	Control mode	0 to FFFFH <div style="display: flex; align-items: center; gap: 5px;"> <span>b15</span> <span>...</span> <span>b10</span> <span>b9</span> <span>b8</span> <span>b7</span> <span>b6</span> <span>b5</span> <span>b4</span> <span>b3</span> <span>b2</span> <span>b1</span> <span>b0</span> </div> 	16-bit unsigned binary	8H	User/system
+3	ALM	Alarm detection	0 to FFFFH <div style="display: flex; align-items: center; gap: 5px;"> <span>b15</span> <span>b14</span> <span>...</span> <span>b0</span> </div>  <p>SPA 0: Loop RUN 1: Loop STOP</p>	16-bit unsigned binary	4000H	User/system
+12 +13	MV	Manipulated value	-10 to 110 [%]	Single-precision real number	0.0	User

## Processing details

This instruction performs output conversion of the manipulated value (MV) in the device specified by (d2), and stores the result in the device specified by (d1).

The following is the processing block diagram of the S.MOUT instruction. (The numbers (1) to (3) in the diagram indicate the order of the processing.)



### ■ Mode determination (1)

The following processing is performed depending on the control mode (MODE).

Control mode (MODE)	Processing details
MAN, CMB, CMV, LCM	<ul style="list-style-type: none"><li>• The manipulated value (MV) is used for the output value (BW).</li><li>• "Output conversion processing (2)" is performed.</li></ul>
AUT, CAB, CAS, CCB, CSV, LCA, LCC	The last output value (BW) is held.

### ■ Output conversion processing (2)

The output value (BW) is calculated from the following expression.

$$BW = \frac{NMAX - NMIN}{100} \times MV + NMIN$$

### ■ Loop stop processing (3)

The following processing is performed according to the SPA status of the alarm detection (ALM).

SPA status	Processing details
0	Performs "Mode determination (1)".
1	Performs the following operations and terminates the S.MOUT instruction. <ul style="list-style-type: none"><li>• The last output value (BW) is held.</li><li>• The control mode (MODE) is set to MAN.</li></ul>

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s2) or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# Time proportioning

## S.DUTY

This instruction outputs ON and OFF by changing the ON/OFF ratio in a given cycle in proportion to the input data (0 to 100%).

Ladder	ST
	<pre>ENO:=S_DUTY(EN,s1,s2,d1,d2);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.DUTY	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit


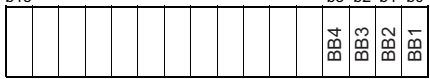
#### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□\□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	—	—	—	—	—	—	—	—	○	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—

#### Input data

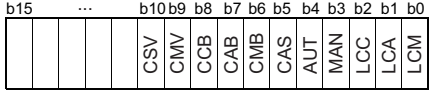
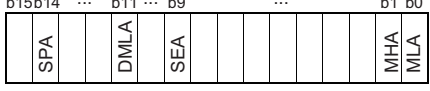
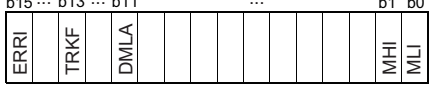
Operand: (s1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value	-999999 to 999999 [%]	Single-precision real number	—	User
+1		(ΔMV)				

## ■Block memory

Operand: (d1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	BW BW 1	Output bit	b15 ... b0  0: Off 1: ON	16-bit unsigned binary	—	System
+1	BB BB1 BB2 BB3 BB4	Alarm Output upper limit alarm Output lower limit alarm Output variation rate alarm	b15 ... b3 b2 b1 b0  0: No alarm 1: Alarm	16-bit unsigned binary	—	System

## ■Loop tag memory

The loop tag memory occupies 96 words from the specified start device.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+1	MODE	Control mode	0 to FFFFH b15 ... b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 	16-bit unsigned binary	8H	User/system
+3	ALM	Alarm detection	0 to FFFFH b15 b14 ... b11 ... b9 ... b1 b0  SPA 0: Loop RUN 1: Loop STOP DMLA, SEA, MHA, MLA 0: No alarm 1: Alarm	16-bit unsigned binary	4000H	User/system
+4	INH	Disable alarm detection	0 to FFFFH b15 ... b13 ... b11 ... b1 b0  TRKF 0: Tracking not set 1: Tracking set ERRI, DMLI, MHI, MLI 0: Alarm detection enabled 1: Alarm detection disabled	16-bit unsigned binary	0H	User/system
+12 +13	MV	Manipulated value	-10 to 110 [%]	Single-precision real number	0.0	User/system
+18 +19	MH	Output upper limit value	-10 to 110 [%]	Single-precision real number	100.0	User
+20 +21	ML	Output lower limit value	-10 to 110 [%]	Single-precision real number	0.0	User
+48 +49	DML	Output variation rate limit value	0 to 100 [%]	Single-precision real number	100.0	User
+54 +55	I	Integral constant	0 to 999999 [s]	Single-precision real number	10.0	User
+62 +63	MVP	MV internal operation value	-999999 to 999999 [%]	Single-precision real number	0.0	System

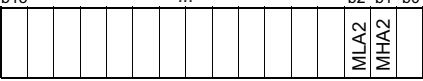
Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+68 +69	CTDUTY	Control output cycle	0 to 999999 [s] Set a value within the following range. $\frac{CTDUTY}{\Delta T} \leq 32767$	Single-precision real number	1.0	User

### Loop tag past value memory

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

The loop tag past value memory occupies 32 words after the loop tag memory.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+116	ALM2	Alarm detection 2	b15 ... b2 b1 b0  MHA2, MLA2 0: No alarm 1: Alarm	—	—	System
+118	—	Control output cycle counter initialization completion flag	—			
+119	—	Control output cycle counter (The value is rounded off to the nearest whole number.)	—			
+120	—	Output counter	—			
+121	—	Output ON counter	—			

### Execution cycle (ΔT)

Set the execution cycle (ΔT) in SD816 and SD817 in single-precision real number.

### Processing details

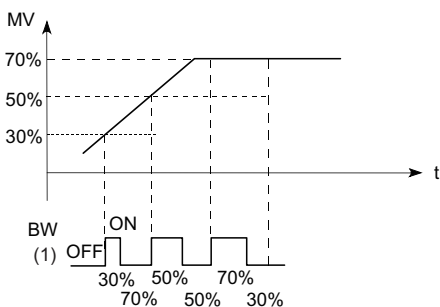
This instruction calculates the manipulated value (MV) from the input value (E1=ΔMV) in the device specified by (s1) by performing input addition processing.

The instruction also turns ON or OFF the device specified by (d1) in proportion to the manipulated value (MV).

The ON/OFF time is a value determined by assuming the time specified by the control output cycle (CTDUTY) as 100%.

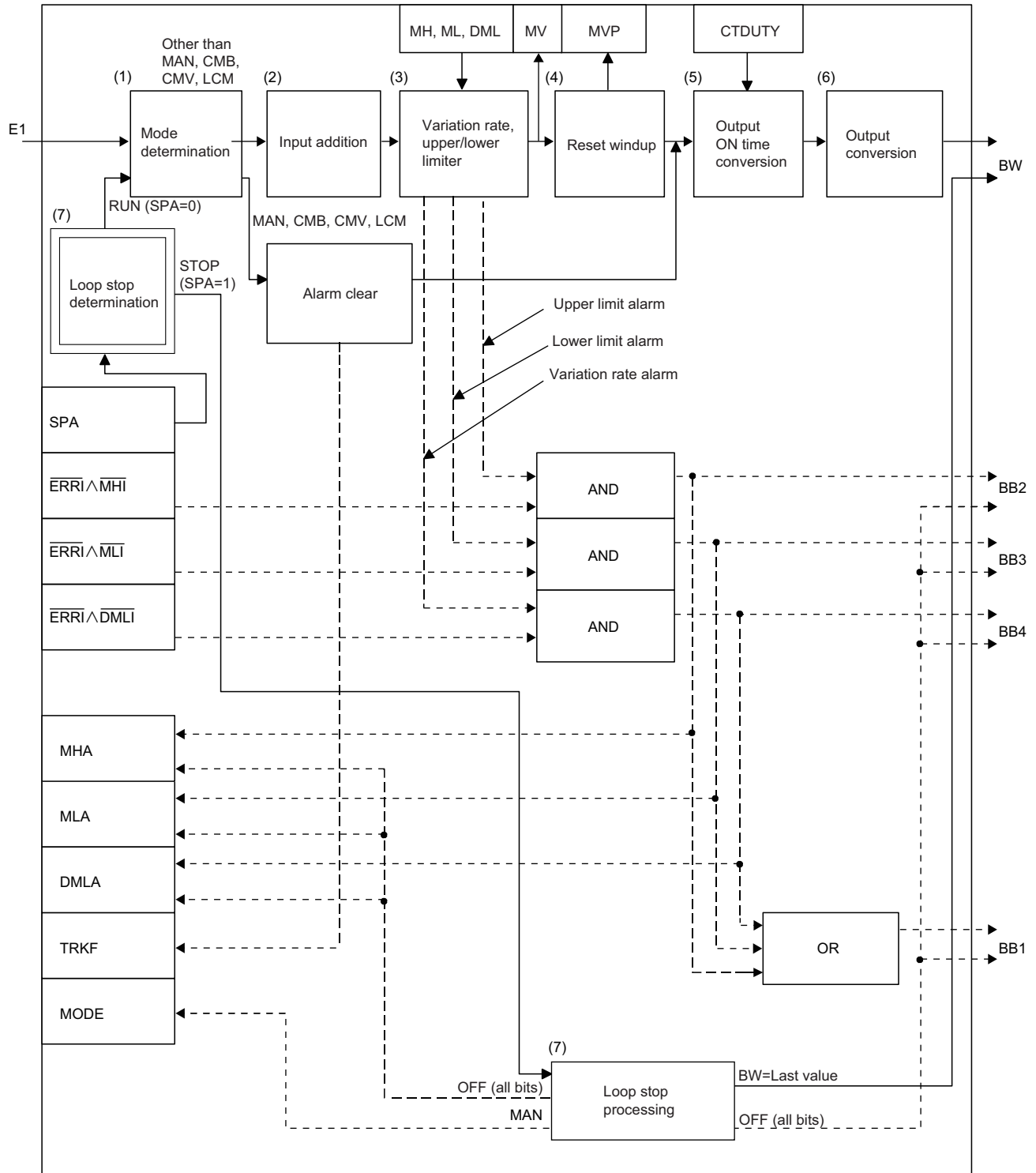
The ON/OFF time is switched every execution cycle.

The instruction also performs variation rate & upper/lower limiter and reset windup of the calculated manipulated value (MV).



(1) Device specified by (d1)

The following is the processing block diagram of the S.DUTY instruction. (The numbers (1) to (7) in the diagram indicate the order of the processing.)





## ■ Mode determination (1)

The following processing is performed depending on the control mode (MODE).

Control mode (MODE)	Processing details
MAN, CMB, CMV, LCM (alarm clear processing)	<ul style="list-style-type: none"> <li>The MHA, MLA, and DMLA of the alarm detection (ALM) are set to 0.</li> <li>The MHA2 and MLA2 of the alarm detection 2 (ALM2) are set to 0.</li> <li>The alarm bits (BB1, BB2, BB3, and BB4) are set to 0.</li> <li>TRKF of INH is set to 1.</li> <li>"Output ON time conversion processing (5)" is performed.</li> </ul>
AUT, CAB, CAS, CCB, CSV, LCA, LCC	"Input addition processing (2)" is performed. However, when ALM SEA is 1 and SM817 is on, alarm bits BB1, BB2, BB3, and BB4 are set to 0 and the S.DUTY instruction is terminated.

## ■ Input addition processing (2)

A temporary MV(T) is calculated on the basis of the input value ( $E1=\Delta MV$ ). The following processing is performed depending on the TRKF of INH.

Tracking flag (TRKF) status	Processing details
1	<ul style="list-style-type: none"> <li>The manipulated value (MV) is stored in the MV internal operation value (MVP).</li> <li>The input value (E1) is set to 0. (<math>\Delta MV=0</math>)</li> <li>TRKF of INH is set to 0.</li> <li>A temporary MV(T) is calculated according to the following expression.  <math>T=E1+MVP</math>  <math>MVP=T</math> </li> </ul>
0	A temporary MV(T) is calculated according to the following expression. $T=E1+MVP$ $MVP=T$

## ■ Variation rate & upper/lower limiter (3)

Variation rates and upper/lower limits are checked for the difference between the temporary MV(T) and manipulated value (MV), and the data after limiter processing and an alarm are output.

- Variation rate limiter processing performs the following operations, and outputs the result to the output variation rate alarm (BB4) and the DMLA of the alarm detection (ALM).

Condition	BB4, DMLA	Result (T1)
$ T-MV  \leq DML$	0	T
$(T - MV) > DML$	1 <sup>*1</sup>	MV + DML
$(T - MV) < -DML$	1 <sup>*1</sup>	MV - DML

\*1 If the DMLI or ERRI of the disable alarm detection (INH) is set to 1, the output variation rate alarm (BB4) and the DMLA of the alarm detection (ALM) are set to 0.

- Upper/lower limiter processing performs the following operations, and outputs the result to the output upper limit alarm (BB2); output lower limit alarm (BB3); MHA and MLA of the alarm detection (ALM); and MHA2 and MLA2 of the alarm detection (ALM2).

Condition	BB3, MLA, MLA2	BB2, MHA, MHA2	MV
$T1 > MH$	0	1 <sup>*2</sup>	MH
$T1 < ML$	1 <sup>*3</sup>	0	ML
$ML \leq T1 \leq MH$	0	0	T1

\*2 If the MHI or ERRI of the disable alarm detection (INH) is set to 1, the output upper limit alarm (BB2) and the MHA of the alarm detection (ALM) are set to 0.

Note that the MHA2 of the alarm detection 2 (ALM2) remains 1.

\*3 If the MLI or ERRI of the disable alarm detection (INH) is set to 1, the output lower limit alarm (BB3) and the MLA of the alarm detection (ALM) are set to 0.

Note that the MLA2 of the alarm detection 2 (ALM2) remains 1.

## ■Reset windup (4)

If the manipulated value (MV) goes beyond the upper or lower limit, the following operation is performed to return it to the upper or lower limit and enable immediate response when the deviation is inverted. However, when the integral constant (I) is 0, reset windup processing is not performed.

Condition	Operational expression
$T1 > MH, \frac{\Delta T}{I} \leq 1$	$MVP = (\frac{\Delta T}{I})(MH - T) + T$
$T1 < ML, \frac{\Delta T}{I} \leq 1$	$MVP = (\frac{\Delta T}{I})(ML - T) + T$

## ■Output ON time conversion processing (5)

The following processing is performed by output ON time conversion processing.

Condition	Processing details
The control output cycle (CTDUTY) has been reached.	The output ON counter is calculated using the following expression. The output counter is cleared to 0 at this time. $\frac{CTDUTY}{\Delta T} \times MV \times \frac{1}{100}$ The output ON counter is rounded off to the nearest whole number.
The control output cycle (CTDUTY) has not been reached.	The output counter is incremented by, 1 and "output conversion processing (6)" is performed.

## ■Output conversion processing (6)

The following processing is performed by output conversion processing.

Condition	BW1
Output counter < Output ON counter	1
Output counter ≥ Output ON counter	0

## ■Loop stop processing (7)

The following processing is performed according to the SPA status of the alarm detection (ALM).

SPA status	Processing details
1	The loop stops. When the loop stops, the following operations are performed and the S.DUTY instruction ends. <ul style="list-style-type: none"> <li>The output bit (BW1) is output at the last ON/OFF rate.</li> <li>The DMLA, MHA, and DLA of the alarm detection (ALM) are set to 0.</li> <li>The MHA2 and MLA2 of the alarm detection 2 (ALM2) are set to 0.</li> <li>The control mode (MODE) is set to MAN.</li> <li>The alarm bits (BB1, BB2, BB3, and BB4) are set to 0.</li> </ul>
0	The loop runs and "mode determination (1)" is performed.

## ■Hold processing (8)

This processing specifies whether to hold the output value by the S.DUTY instruction when a sensor error occurs (detected by the S.IN instruction). The hold processing is performed when the value is determined as RUN by "Loop Stop Determination". SM817 is used to specify whether to hold the manipulated value (MV) when a sensor alarm occurs.

- SM817 = OFF: Do not hold the manipulated value (MV).
- SM817 = ON: Hold the manipulated value (MV).

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1) or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Control output cycle (CTDUTY) < 0 The execution cycle (ΔT) setting is less than 0. (Control output cycle (CTDUTY) ÷ execution cycle (ΔT)) > 32767

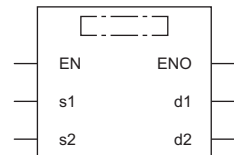
# Batch counter

## S.BC

This instruction compares the input data with the set value, and outputs bit data when it reaches the set value.

Ladder	ST
	<pre>ENO:=S_BC(EN,s1,s2,d1,d2);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.BC	

### Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	32-bit unsigned binary
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□\□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	—	—	—	—	—	—	—	—	○	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—

### Input data

Operand: (s1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value	0 to 2147483647	32-bit unsigned binary	—	User
+1						

## ■Block memory

Operand: (d1)							
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by	
+0	BW	BW 1	Output 1	<p>0: Off 1: ON</p>	16-bit unsigned binary	—	System
		BW 2	Output 2				
+1	BB	BB1	Alarm	<p>0: No alarm 1: Alarm</p>	16-bit unsigned binary	—	System
		BB2	Upper limit alarm				
		BB3	Variation rate alarm				

## ■Loop tag memory

The loop tag memory occupies 96 words from the specified start device.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+3	ALM	Alarm detection	0 to FFFFH <p>PHA, DPPA 0: No alarm 1: Alarm</p>	16-bit unsigned binary	4000H	User/ system
+4	INH	Disable alarm detection	0 to FFFFH <p>0: Alarm detection enabled 1: Alarm detection disabled</p>	16-bit unsigned binary	0H	User/ system
+14 +15	SV1	Set value 1	0 to 2147483647	32-bit unsigned binary	0	User
+16 +17	SV2	Set value 2	0 to 2147483647	32-bit unsigned binary	0	User
+26 +27	PH	Upper limit alarm value	0 to 2147483647	32-bit unsigned binary	0	User
+42 +43	CTIM	Variation rate alarm check time	0 to 999999 [s] Set a value within the following range. $\frac{CTIM}{\Delta T} \leq 32767$	Single-precision real number	0.0	User
+44 +45	DPL	Variation rate alarm value	0 to 2147483647	32-bit unsigned binary	0	User

## ■Loop tag past value memory

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

The loop tag past value memory occupies 32 words after the loop tag memory.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+124	—	Variation rate monitoring counter initialization completion flag	—	—	—	System
+125	—	Variation rate monitoring counter (rounded off to the nearest whole number)	—	—	—	System
+126 +127	$X_{n-m}$	—	—	—	—	System

## ■Execution cycle ( $\Delta T$ )

Set the execution cycle ( $\Delta T$ ) in SD816 and SD817 in single-precision real number.

### Processing details

This instruction compares the input value (E1) with SV1/SV2 and outputs bit data when E1 reaches SV1/SV2.

The instruction also performs upper limit check, variation rate check, and output conversion processing of the input value (E1) at that time.

## ■Upper limit check (1)

The upper limit check performs the following operations, and outputs the result to the upper limit alarm (BB2) and PHA of the alarm detection (ALM).

Condition	BB2, PHA
$E1 > PH$	1*1
Others	0

\*1 If PHI or ERR1 of the disable alarm detection (INH) is set to 1, the upper limit alarm (BB2) and the PHA of the alarm detection (ALM) are set to 0.

## ■Variation rate check processing (2)

A variation rate alarm check is performed during the variation rate alarm check time (CTIM) in the device specified by (d2).

For the variation rate alarm check, the variation of the input value (E1) is compared with the variation rate alarm value (DPL) every execution cycle ( $\Delta T$ ).

Condition	BB3, DPPA
$(X_n - X_{n-m}) \geq DPL$	1*1
Others	0

\*1 If the DPP1 or ERR1 of the disable alarm detection (INH) is set to 1, the variation rate alarm (BB3) and the DPPA of the alarm detection (ALM) are set to 0.

The variation rate alarm counter is calculated using the following expression.

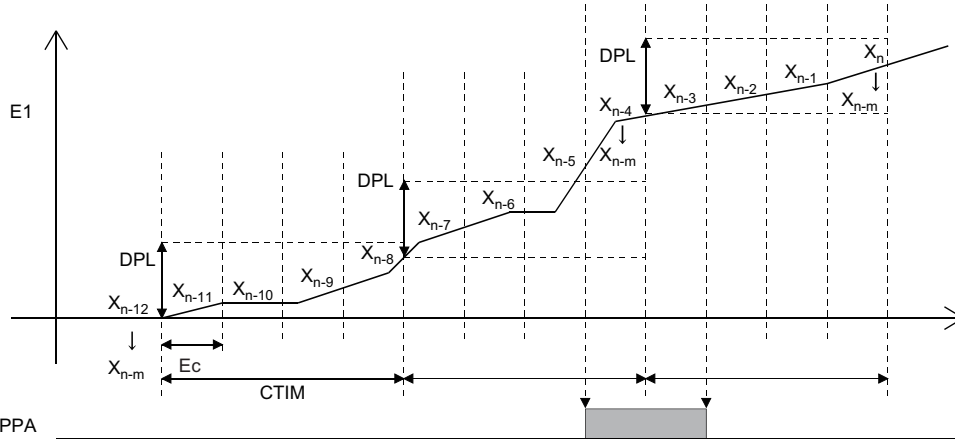
$$m = \frac{CTIM}{\Delta T}$$

Set CTIM and  $\Delta T$  so that the variation rate alarm counter ( $m \geq 2$ ).

No processing is performed when variation rate alarm counter ( $m = 0$ ).

**Ex.**

When variation rate alarm counter ( $m = 4$ ), operations are performed as shown below.



BB3, DPPA

Ec: Execution cycle

### ■ Output conversion processing (3)

In output conversion processing, the following operations are performed and the result is stored in output 1 (BW1)/output 2 (BW2).

Condition	BW1	BW2
$E1 < 0$	0	0
$0 \leq E1 < SV1$	0	—
$E1 \geq SV1$	1	—
$0 \leq E1 < SV2$	—	0
$E1 \geq SV2$	—	1

### Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1) or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Variation rate alarm check time (CTIM) < 0
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	(Variation rate alarm check time (CTIM) ÷ execution cycle ( $\Delta T$ )) > 32767

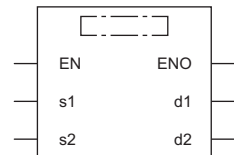
# Pulse integration

## S.PSUM

This instruction integrates and outputs the number of input pulses.

Ladder	ST
	<pre>ENO:=S_PSUM(EN,s1,s2,d1,d2);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.PSUM	

### Setting data


#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

#### Applicable devices

Operand	Bit		Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□\□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	○	—	—	—	—	
(d2)	—	—	○	—	—	—	○	—	—	—	—	

## Input data

Operand: (s1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Input value	Use a ring counter consisting of 16 bits or more. <ul style="list-style-type: none"> <li>16-bit ring counter 00000000H → 0000FFFFH → 00000000H[pulse]</li> <li>24-bit ring counter 00000000H → 00FFFFFFH → 00000000H[pulse]</li> <li>32-bit ring counter 00000000H → FFFFFFFFH → 00000000H[pulse]</li> </ul> However, set 32767 (7FFFH) or less for the pulse increment in each instruction execution.	32-bit unsigned binary	—	User
+2	e	e1 Integration start signal e2 Integration hold signal	 <p>Integration start signal 0: Integration stop/reset 1: Integration start Integration hold signal 0: Integration hold clear 1: Integration hold</p>	16-bit unsigned binary	—	User

## Block memory

Operand: (d1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW1	Output value (integral part)	0 to 2147483647	32-bit unsigned binary	—	System
+2 +3	BW2	Output value (decimal part)	0 to 2147483647			

## Operation constant

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	W	Weight per pulse	1 to 999	16-bit unsigned binary	1	User
+1	U	Unit conversion constant	1, 10, 100, 1000	16-bit unsigned binary	1	User
+2 +3	HILMT	Integration upper limit	1 to 2147483647	32-bit unsigned binary	2147483647	User
+4	SUMPTN	Integration pattern	<ul style="list-style-type: none"> <li>0: Return to 0 when the integration upper limit (HILMIT) is exceeded.</li> <li>1: Holds the integration upper limit (HILMIT) when it is exceeded.</li> </ul>	16-bit unsigned binary	0	User

## Loop tag memory

The loop tag memory occupies 96 words from the specified start device.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+10 +11	SUM1	Integrated value (integral part)	0 to 2147483647	32-bit unsigned binary	0	System
+12 +13	SUM2	Integrated value (decimal part)	0 to 2147483647	32-bit unsigned binary	0	System



## Loop tag past value memory

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

The loop tag past value memory occupies 32 words after the loop tag memory.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+116 +117	E1 <sub>n-1</sub>	Last input value	—	—	—	System

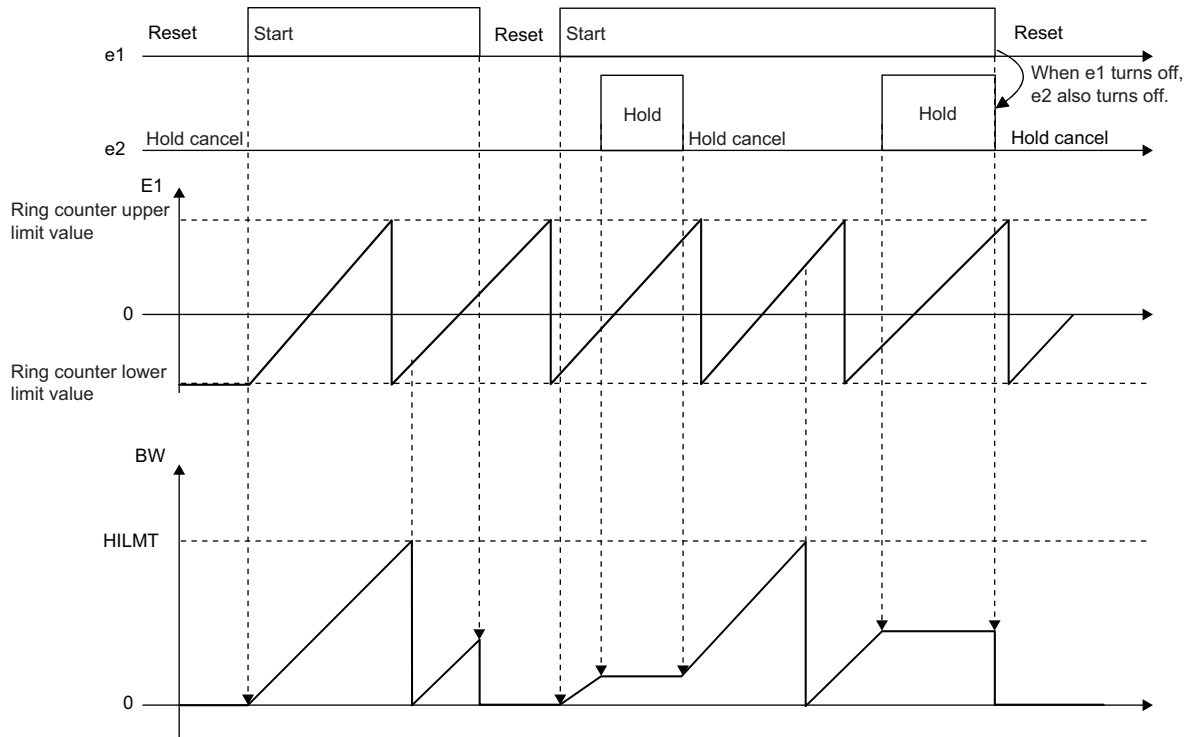
## Processing details

This instruction integrates the input value (E1) in the device specified by (s1), and stores the result in the device specified by (d1).

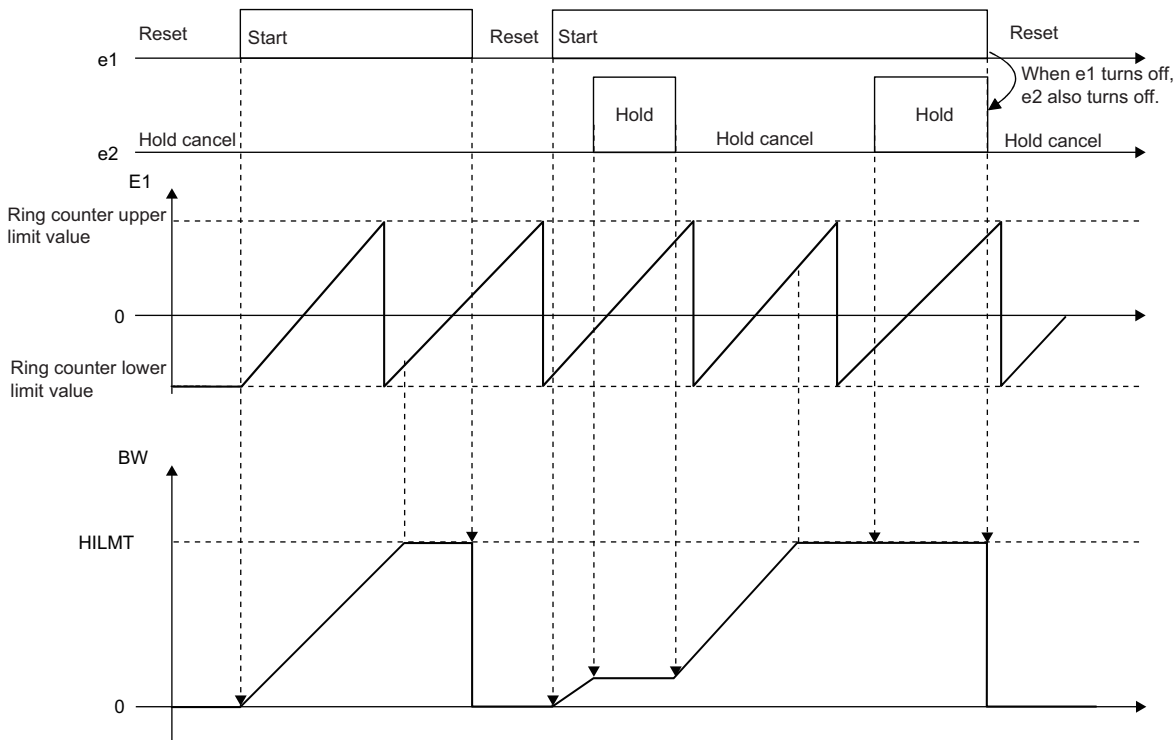
HILMT and SUMPTN can be used to specify whether to return SUM1/SUM2 to 0 or hold the HILMT value when BW1/BW2 exceeds HILMT.

e1 and e2 can be used to start or stop integration of E1.

- Operation performed when the integration pattern is set to "Return to 0 when HILMT is exceeded"



- Operation performed when the integration pattern is set to "Hold the HILMT value when HILMT is exceeded"



### Input value increment operation processing

In input value increment operation processing, the following processing is performed for the input value (E1).

e1	e2	Input value increment (T1)
0	0	—
0	1	—
1	0	$E1 - E1_{n-1}$
1	1	—

### Integration value calculation processing

In integrated value calculation processing, the following processing is performed for the input value increment (T1).

e1	e2	Integrated value (decimal part) (T2), integrated value (decimal part) (T3)
0	0	T2 = 0 T3 = 0
0	1	T2 = 0 T3 = $0^{*1}$
1	0	T4 = $\{(T1 \times W) \div U\}$ quotient <integral part> T5 = $\{(T1 \times W) \div U\}$ remainder <decimal part> T2 = SUM1 + T4 + $\{((SUM2 + T5) \div U)\}$ quotient <integral part> T3 = $\{((SUM2 + T5) \div U)\}$ remainder <decimal part>
1	1	T2 = SUM1 T3 = SUM2

\*1 In the case of integration stop/reset (e1 = 0), processing is performed by assuming it as integration hold clear (e2 = 0).

### Output conversion processing

In output conversion processing, the following processing is performed for the integrated values (T2, T3).

SUMPTN	Condition	BW1, SUM1	BW2, SUM2
0	T2 ≥ HILMT	BW1 = T2 ÷ remainder of HILMT SUM1 = T2 ÷ remainder of HILMT	BW2 = T3 SUM2 = T3
	Others	BW1 = T2 SUM1 = T2	BW2 = T3 SUM2 = T3
1	T2 ≥ HILMT	BW1 = HILMT SUM1 = HILMT	BW2 = 0 SUM2 = 0
	Others	BW1 = T2 SUM1 = T2	BW2 = T3 SUM2 = T3

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.

# 21.4 Control Operation Instructions

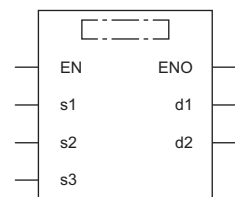
## Basic PID control

### S.PID

This instruction performs process value differential type (inexact differential) PID operation. The instruction performs the following processing steps: SV setting, tracking, gain (Kp) operation, PID operation, and deviation check.

Ladder	ST
	<pre>ENO:=S_PID(EN,s1,s2,s3,d1,d2);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.PID	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
(s3)	When E2 is used: Set value start device When E2 is not used: Dummy device	Refer to "Set value".	Single-precision real number
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

#### Applicable devices


Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□, U3E□\(\H)G□	Z	LT, LST, LC	LZ		K	H	E	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	—	○	—	—	—	—	○	—	—	—	—

## Input data

Operand: (s1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Input value	-999999 to 999999	Single-precision real number	—	User

## Block memory

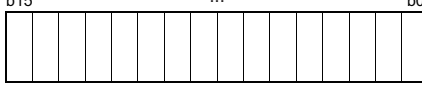
Operand: (d1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value ( $\Delta$ MV)	-999999 to 999999	Single-precision real number	—	System
+2	BB	BB1	Large deviation alarm	16-bit unsigned binary	—	System



0: No alarm  
1: Alarm

## Operation constant

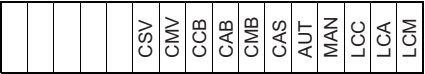
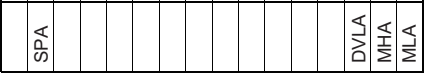

Operand: (s2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	MTD	Derivative gain	0 to 999999	Single-precision real number	8.0	User
+2 +3	DVLS	Large deviation alarm Hysteresis	0 to 100 [%]	Single-precision real number	2.0	User
+4	PN	Operation mode	0: Reverse action 1: Direct action	16-bit unsigned binary	0	User
+5	TRK	Tracking bit	0: Tracking not performed 1: Tracking performed	16-bit unsigned binary	0	User
+6	SVPTN	Set value pattern	0 to 3	16-bit unsigned binary	3	User



(1) Use of set value  
Specify whether to use the set value (E2) or not.  
0: Used  
1: Not used  
(2) Set value pattern  
Specify whether to use the upper loop MV as the set value (E2) or not.  
0: E2 is the upper loop MV.  
1: E2 is not the upper loop MV.

## Loop tag memory

The loop tag memory occupies 96 words from the specified start device.

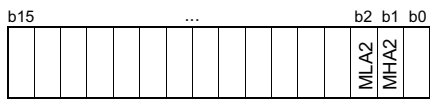
Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+1	MODE	Control mode	0 to FFFFH <div style="display: flex; justify-content: space-between; align-items: center;"> <span>b15</span> <span>...</span> <span>b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0</span> </div> 	16-bit unsigned binary	8H	User/system
+3	ALM	Alarm detection	0 to FFFFH <div style="display: flex; justify-content: space-between; align-items: center;"> <span>b15 b14</span> <span>...</span> <span>b2 b1 b0</span> </div>  <p>SPA            0: Loop RUN            1: Loop STOP            DVLA, MHA, MLA            0: No alarm            1: Alarm</p>	16-bit unsigned binary	4000H	User/system
+4	INH	Disable alarm detection	0 to FFFFH <div style="display: flex; justify-content: space-between; align-items: center;"> <span>b15</span> <span>...</span> <span>b13</span> <span>...</span> <span>b2 b1 b0</span> </div>  <p>TRKF            0: Tracking not set            1: Tracking set            ERRI, DVLI, MHI, MLI            0: Alarm detection enabled            1: Alarm detection disabled</p>	16-bit unsigned binary	0H	User/system
+14 +15	SV	Set value	RL to RH	Single-precision real number	0.0	User
+16 +17	DV	Deviation	-110 to 110 [%]	Single-precision real number	0.0	System
+22 +23	RH	Engineering value upper limit	-999999 to 999999	Single-precision real number	100.0	User
+24 +25	RL	Engineering value lower limit	-999999 to 999999	Single-precision real number	0.0	User
+46 +47	CT	Control cycle	0 to 999999 [s] Set a value within the following range. $\frac{CT}{\Delta T} \leq 32767$	Single-precision real number	1.0	User
+50 +51	DVL	Deviation limit value	0 to 100 [%]	Single-precision real number	100.0	User
+52 +53	P	Gain	0 to 999999	Single-precision real number	1.0	User
+54 +55	I	Integral constant	0 to 999999 [s]	Single-precision real number	10.0	User
+56 +57	D	Derivative constant	0 to 999999 [s]	Single-precision real number	0.0	User
+58 +59	GW	Gap width	0 to 100[%]	Single-precision real number	0.0	User
+60 +61	GG	Gap gain	0 to 999999	Single-precision real number	1.0	User
+62 +63	MVP	MV internal operation value	-999999 to 999999 [%]	Single-precision real number	0.0	System

### Loop tag past value memory

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

The loop tag past value memory occupies 32 words after the loop tag memory.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+96	—	Control cycle counter initialization completion flag				System
+97		Control cycle counter (The value is rounded off to the nearest whole number.)				
+102	$B_{n-1}$	Last value				
+103						
+104	$PV_n$	Process value				
+105						
+106	$PV_{n-1}$	Last process value				
+107						
+108	$PV_{n-2}$	Last-but-one process value				
+109						
+110	$DV_{n-1}$	Last deviation value	 <p>MHA2, MLA2 0: No alarm 1: Alarm</p>			
+111						
+116	ALM2	Alarm detection 2				

### Set value

The set value (E2) is valid only when b0 of the set value pattern (SVPTN) is set to 0 (Used). To use the upper loop MV as the set value (E2), specify the device (offset +12) where the manipulated value (MV) of the upper loop is set.

If E2 is not used, specify a dummy device (SD820).

Operand: (s3)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E2	Set value	-10 to 110 [%]	Single-precision real number	0.0	User
+1						

### Execution cycle ( $\Delta T$ )

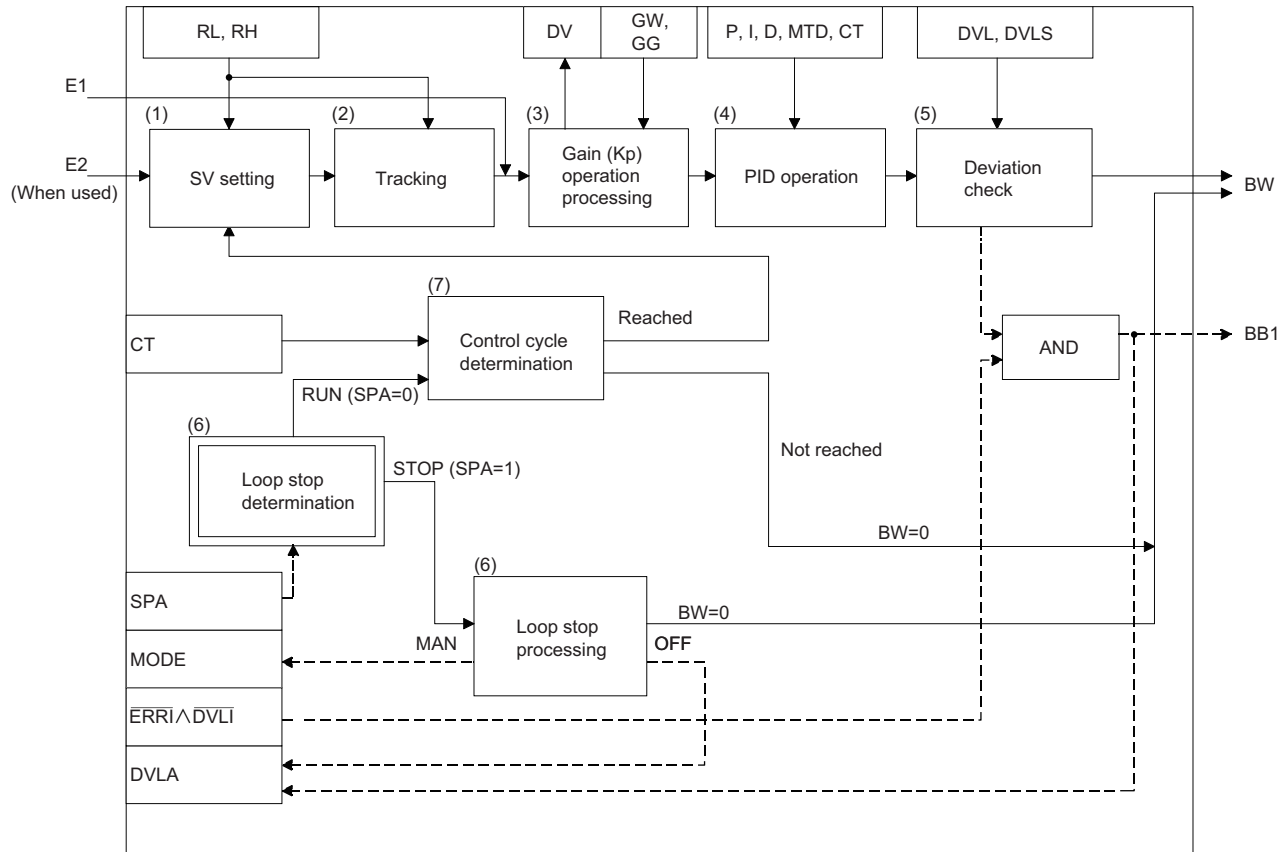
Set the execution cycle ( $\Delta T$ ) in SD816 and SD817 in single-precision real number.

## Processing details

This instruction performs PID operation when the specified control cycle is reached. (The PID operation is of the velocity type/ process value differential type (inexact differential).)

At this time, the instruction also performs the following processing steps: SV setting, tracking, gain (Kp) operation, and deviation check.

The following is the processing block diagram of the S.PID instruction. (The numbers (1) to (7) in the diagram indicate the order of the processing.)



### SV setting processing (1)

The following processing is performed depending on the control mode (MODE) setting.

Control mode (MODE) setting	Processing details
CAS, CCB, CSV	<ul style="list-style-type: none"> <li>If the set value (E2) is specified, engineering value transformation processing (refer to the following expression) is performed, and then "tracking processing (2)" is performed.</li> </ul> $SV_n' = \frac{RH-RL}{100} \times E2 + RL$ <ul style="list-style-type: none"> <li>If the set value (E2) is not specified, "tracking processing (2)" is performed without performing engineering value transformation processing.</li> </ul>
MAN, AUT, CMV, CMB, CAB, LCM, LCA, LCC	"Tracking processing (2)" is performed.

### Tracking processing (2)

- The set value (SV) is inversely transformed from the engineering value and  $SV_n'$  is calculated (refer to the following expression).

$$SV_n' = \frac{100}{RH-RL} \times (SV_n - RL)$$

- Tracking processing is performed when all of the following conditions are satisfied.
  - The tracking bit (TRK) is set to 1.
  - The set value (E2) is used.
  - The control mode (MODE) is set to any of the following: MAN, AUT, CMV, CMB, CAB, LCM, LCA, or LCC.  
 $E2 = SV_n'$
- If the set value (E2) is the upper loop MV, the TRKF of the disable alarm detection (INH) of the upper loop is set to 1.



### Gain (Kp) operation processing (3)

- The deviation (DV) is calculated under the following conditions.

Condition	Operational expression
Direct action (PN = 1)	$DV = E1 - SV_n'$
Reserve action (PN = 0)	$DV = SV_n' - E1$

- The output gain (K) is calculated under the following conditions.

Condition	Operational expression
$ DV  \leq GW$	$K = GG$
$ DV  > GW$	$K = 1 - \frac{(1-GG) \times GW}{ DV }$

### PID operation (4)

The PID operation is performed with the following operational expression.

Item	Operational expression
$B_n$	Direct action (PN = 1) $B_{n-1} + \frac{M_D \times T_D}{M_D \times CT + T_D} \times \left\{ (PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times B_{n-1}}{T_D} \right\}$
	Reserve action (PN = 0) $B_{n-1} + \frac{M_D \times T_D}{M_D \times CT + T_D} \times \left\{ -(PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times B_{n-1}}{T_D} \right\}$
$BW(\Delta MV)$	$K_P \times \left\{ (DV_n - DV_{n-1}) + \frac{CT}{T_I} \times DV_n + B_n \right\}$

$K_P$ :  $K \times$  Gain (P),  $M_D$ : Derivative gain (MTD),  $T_I$ : Integral constant (I),  $T_D$ : Derivative constant (D)

Note that special processing is performed in the following cases.

Condition	Processing
In either of the following cases: 1. Derivative constant (D) = 0 ( $T_D = 0$ ) 2. Control mode (MODE) = MAN, LCM, or CMV	$B_n = 0$ (Note that the loop tag past value memory is set.)
In any of the following cases: 1. Integral constant (I) = 0 ( $T_I = 0$ ) 2. MHA2 or MLA2 of alarm detection 2 (ALM2) is 1.  MVP > MH  and $\frac{CT}{T_I} \times DV_n > 0$  3. MHA2 or MLA2 of alarm detection 2 (ALM2) is 1.  MVP < ML  and $\frac{CT}{T_I} \times DV_n < 0$	$\frac{CT}{T_I} \times DV_n = 0$

### Deviation check (5)

A deviation is checked under the following conditions, and the result is output to the DVLA of the alarm detection (ALM) and the large deviation alarm (BB1).

Condition	Result
$DVL <  DV $	$DVLA = BB1 = 1^{*1}$
$(DVL - DVLS) <  DV  \leq DVL$	$DVLA = BB1 = \text{Last value status hold}^{*1}$
$ DV  \leq (DVL - DVLS)$	$DVLA = BB1 = 0$

\*1 If the DML1 or ERR1 of the disable alarm detection (INH) is set to 1, the DVLA of the alarm detection (ALM) and the large deviation alarm (BB1) are set to 0.

## ■ Loop stop processing (6)

The following processing is performed according to the SPA status of the alarm detection (ALM).

SPA status	Processing details
1	The loop stops. When the loop stops, the following operations are performed and the S.PID instruction ends. <ul style="list-style-type: none"><li>• The output value (BW (<math>\Delta MV</math>)) is set to 0.</li><li>• The DVLA of alarm detection (ALM) is set to 0.</li><li>• The control mode (MODE) is set to MAN.</li><li>• BB1 of BB is set to 0.</li></ul>
0	The loop runs and "control cycle determination processing (7)" is performed.

## ■ Control cycle determination (7)

If the specified control cycle is not reached, BW ( $\Delta MV$ ) is set to 0 and the S.PID instruction is terminated.

If the specified control cycle is reached, "SV setting processing (1)" is performed.

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1), (s2), or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.

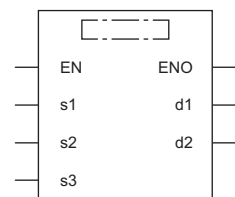
# Two-degree-of-freedom PID control

## S.2PID

This instruction performs two-degree-of-freedom PID control operation (inexact differential). The instruction performs the following processing steps: SV setting, tracking, gain (Kp) operation, two-degree-of-freedom PID control operation, and deviation check.

Ladder	ST
	ENO:=S_2PID(EN,s1,s2,s3,d1,d2);

### FBD/LD



### ■ Execution condition

Instruction	Execution condition
S.2PID	

### Setting data

#### ■ Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
(s3)	When E2 is used: Set value start device When E2 is not used: Dummy device	Refer to "Set value".	Single-precision real number
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

#### ■ Applicable devices


Operand	Bit		Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□, U3E□\(\H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	○	—	—	—	—	
(d2)	—	—	○	—	—	—	○	—	—	—	—	
(s3)	—	—	○	—	—	—	○	—	—	—	—	

## Input data

Operand: (s1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Input value	-999999 to 999999 [%]	Single-precision real number	—	User

## Block memory

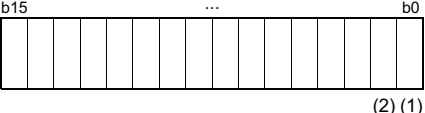
Operand: (d1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value ( $\Delta$ MV)	-999999 to 999999 [%]	Single-precision real number	—	System
+2	BB	BB1	Large deviation alarm	16-bit unsigned binary	—	System



0: No alarm  
1: Alarm

## Operation constant

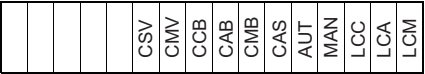
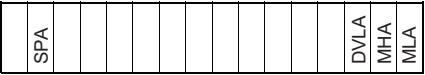
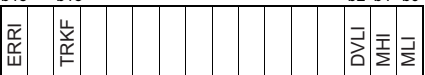
Operand: (s2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	MTD	Derivative gain	0 to 999999	Single-precision real number	8.0	User
+2 +3	DVLS	Large deviation alarm hysteresis	0 to 100 [%]	Single-precision real number	2.0	User
+4	PN	Operation mode	0: Reverse action 1: Direct action	16-bit unsigned binary	0	User
+5	TRK	Tracking bit	0: Tracking not performed 1: Tracking performed	16-bit unsigned binary	0	User
+6	SVPTN	Set value pattern	0 to 3	16-bit unsigned binary	3	User



(1) Use of set value  
Specify whether to use the set value (E2) or not.  
0: Used  
1: Not used  
(2) Set value pattern  
Specify whether to use the upper loop MV as the set value (E2) or not.  
0: E2 is the upper loop MV.  
1: E2 is not the upper loop MV.

## Loop tag memory

The loop tag memory occupies 96 words from the specified start device.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+1	MODE	Control mode	0 to FFFFH b15 ... b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 	16-bit unsigned binary	8H	User/system
+3	ALM	Alarm detection	0 to FFFFH b15 b14 ... b2 b1 b0  <p>SPA            0: Loop RUN            1: Loop STOP            DVLA, MHA, MLA            0: No alarm            1: Alarm</p>	16-bit unsigned binary	4000H	User/system
+4	INH	Disable alarm detection	0 to FFFFH b15 ... b13 ... b2 b1 b0  <p>TRKF            0: Tracking not set            1: Tracking set            ERRI, DVLI, MHI, MLI            0: Alarm detection enabled            1: Alarm detection disabled</p>	16-bit unsigned binary	0H	User/system
+14 +15	SV	Set value	RL to RH	Single-precision real number	0.0	User
+16 +17	DV	Deviation	-110 to 110 [%]	Single-precision real number	0.0	System
+22 +23	RH	Engineering value upper limit	-999999 to 999999	Single-precision real number	100.0	User
+24 +25	RL	Engineering value lower limit	-999999 to 999999	Single-precision real number	0.0	User
+46 +47	CT	Control cycle	0 to 999999 [s] Set a value within the following range. $\frac{CT}{\Delta T} \leq 32767$	Single-precision real number	1.0	User
+50 +51	DVL	Deviation limit value	0 to 100 [%]	Single-precision real number	100.0	User
+52 +53	P	Gain	0 to 999999	Single-precision real number	1.0	User
+54 +55	I	Integral constant	0 to 999999 [s]	Single-precision real number	10.0	User
+56 +57	D	Derivative constant	0 to 999999 [s]	Single-precision real number	0.0	User
+58 +59	GW	Gap width	0 to 100 [%]	Single-precision real number	0.0	User
+60 +61	GG	Gap gain	0 to 999999	Single-precision real number	1.0	User
+62 +63	MVP	MV internal operation value	-999999 to 999999 [%]	Single-precision real number	0.0	System

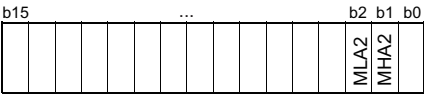
Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+64 +65	$\alpha$	Two-degree-offreedom parameter $\alpha$	0 to 1 Increasing $\alpha$ decreases the manipulated value variation relative to the set value change. (It will take time to stabilize.) Decreasing $\alpha$ increases the manipulated value variation relative to the set value change. However, it strengthens the compensation operation and accordingly makes hunting greater.	Single-precision real number	0.0	User
+66 +67	$\beta$	Two-degree-offreedom parameter $\beta$	0 to 1 Increasing $\beta$ decreases the effect of derivative control on the set value change. Decreasing $\beta$ increases the effect of derivative control on the set value change.	Single-precision real number	1.0	User

### Loop tag past value memory

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

The loop tag past value memory occupies 32 words after the loop tag memory.

Operand: (d2)							
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by	
+96	—	Control cycle counter initialization completion flag	—	—	—	System	
+97		Control cycle counter (The value is rounded off to the nearest whole number.)					
+102 +103	$B_{n-1}$	Last value					
+104 +105	$PV_n$	Process value					
+106 +107	$PV_{n-1}$	Last process value					
+108 +109	$PV_{n-2}$	Last-but-one process value					
+110 +111	$DV_{n-1}$	Last deviation value					
+112 +113	$DV_{n-2}$	Last but one deviation value					
+114 +115	$D_{n-1}$	Last value					
+116	ALM2	Alarm detection 2					 <p>MHA2, MLA2 0: No alarm 1: Alarm</p>

### Set value

The set value (E2) is valid only when b0 of the set value pattern (SVPTN) is set to 0 (Used). To use the upper loop MV as the set value (E2), specify the device (offset +12) where the manipulated value (MV) of the upper loop is set.

If E2 is not used, specify a dummy device (SD820).

Operand: (s3)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E2	Set value	-10 to 110 [%]	Single-precision real number	0.0	User

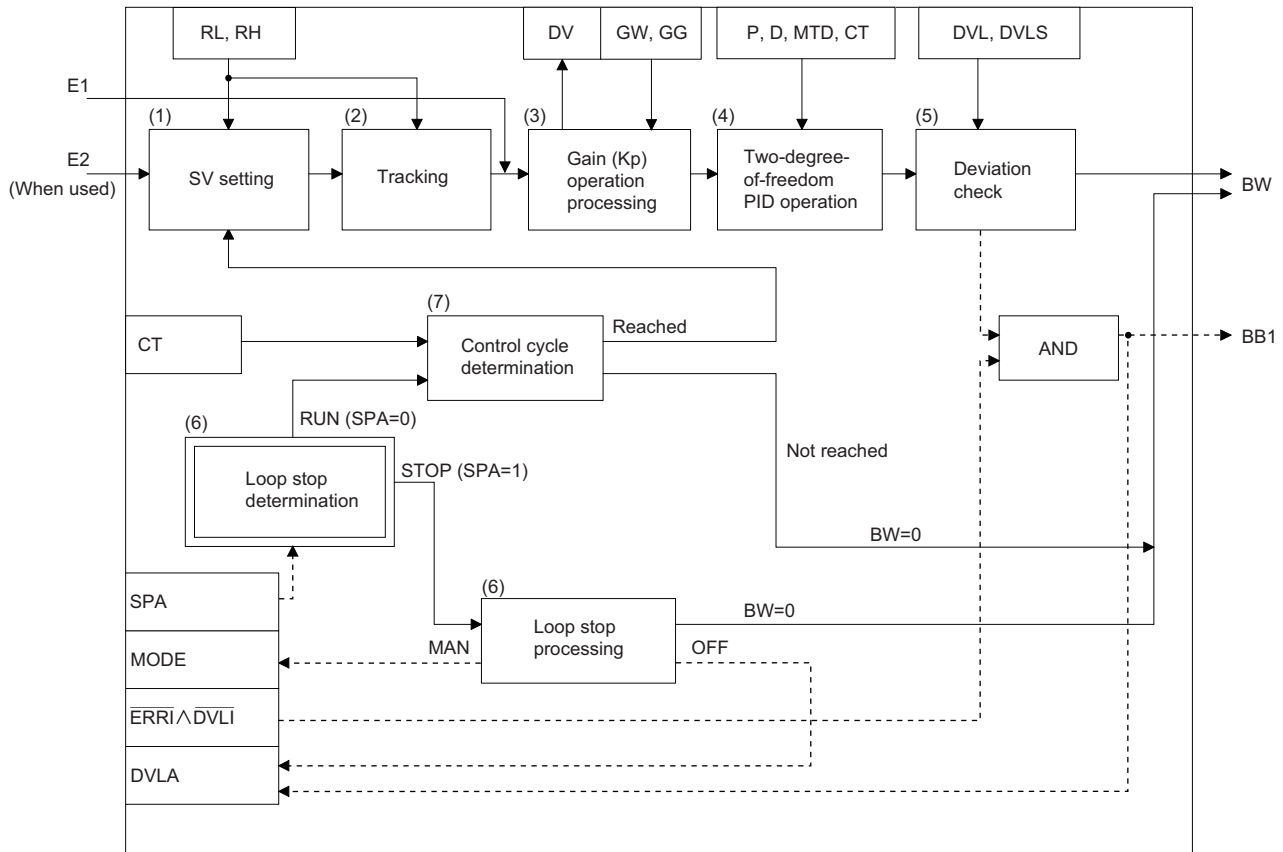
### Execution cycle ( $\Delta T$ )

Set the execution cycle ( $\Delta T$ ) in SD816 and SD817 in single-precision real number.

## Processing details

This instruction performs two-degree-of-freedom PID control operation when the specified control cycle is reached. At this time, the instruction also performs the following processing steps: SV setting, tracking, gain (Kp) operation, and deviation check.

The following is the processing block diagram of the S.2PID instruction. (The numbers (1) to (7) in the diagram indicate the order of the processing.)



### SV setting processing (1)

The following processing is performed depending on the control mode (MODE) setting.

Control mode (MODE) setting	Processing details
CAS, CCB, CSV	<ul style="list-style-type: none"> <li>If the set value (E2) is specified, engineering value transformation processing (refer to the following expression) is performed, and then "tracking processing (2)" is performed.</li> </ul> $SV_n = \frac{RH-RL}{100} \times E2 + RL$ <ul style="list-style-type: none"> <li>If the set value (E2) is not specified, "tracking processing (2)" is performed without performing engineering value transformation processing.</li> </ul>
MAN, AUT, CMV, CMB, CAB, LCM, LCA, LCC	"Tracking processing (2)" is performed.

### Tracking processing (2)

- The set value (SV) is inversely transformed from the engineering value and  $SV_n'$  is calculated (refer to the following expression).

$$SV_n' = \frac{100}{RH-RL} \times (SV_n - RL)$$

- Tracking processing is performed when all of the following conditions are satisfied.
  - The tracking bit (TRK) is set to 1.
  - The set value (E2) is used.
  - The control mode (MODE) is set to any of the following: MAN, AUT, CMV, CMB, CAB, LCM, LCA, or LCC.
 
$$E2 = SV_n'$$
- If the set value (E2) is the upper loop MV, the TRKF of the disable alarm detection (INH) of the upper loop is set to 1.

### ■ Gain (Kp) operation processing (3)

- The deviation (DV) is calculated under the following conditions.

Condition	Operational expression
Direct action (PN = 1)	$DV = E1 - SV_n'$
Reserve action (PN = 0)	$DV = SV_n' - E1$

- The output gain (K) is calculated under the following conditions.

Condition	Operational expression
$ DV  \leq GW$	$K = GG$
$ DV  > GW$	$K = 1 - \frac{(1-GG) \times GW}{ DV }$

### ■ Two-degree-of-freedom PID control (4)

The two-degree-of-freedom PID control operation is performed with the following operational expression.

Item	Operational expression	
$B_n$	$B_{n-1} + \frac{M_D \times T_D}{M_D \times CT + T_D} \times \{ (DV_{n-2} - 2DV_{n-1} + DV_{n-2}) - \frac{CT \times B_{n-1}}{TD} \}$	
$C_n$	Direct action (PN = 1)	$PV_n - PV_{n-1}$
	Reserve action (PN = 0)	$-(PV_n - PV_{n-1})$
$D_n$	Direct action (PN = 1)	$D_{n-1} + \frac{M_D \times T_D}{M_D \times CT + T_D} \times \{ (PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times D_{n-1}}{TD} \}$
	Reserve action (PN = 0)	$D_{n-1} + \frac{M_D \times T_D}{M_D \times CT + T_D} \times \{ -(PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times D_{n-1}}{TD} \}$
$BW(\Delta MV)$	$K_P \times \{ (1-\alpha) \times (DV_n - DV_{n-1}) + \frac{CT}{T_I} \times DV_n + (1-\beta) \times B_n + \alpha \times C_n + \beta \times D_n \}$	

$K_P$ : K × Gain (P),  $M_D$ : Derivative gain (MTD),  $T_I$ : Integral constant (I),  $T_D$ : Derivative constant (D)

Note that special processing is performed in the following cases.

Condition	Processing
In either of the following cases: 1. Derivative constant (D) = 0 ( $T_D = 0$ ) 2. Control mode (MODE) = MAN, LCM, or CMV	$B_n = 0, D_n = 0$ (Note that the loop tag past value memory is set.)
In any of the following cases: 1. Integral constant (I) = 0 ( $T_I = 0$ ) 2. MHA2 or MLA2 of alarm detection 2 (ALM2) is 1.  MVP > MH and $\frac{CT}{T_I} \times DV_n > 0$ 3. MHA2 or MLA2 of alarm detection 2 (ALM2) is 1.  MVP < ML and $\frac{CT}{T_I} \times DV_n < 0$	$\frac{CT}{T_I} \times DV_n = 0$



### ■Deviation check (5)

A deviation is checked under the following conditions, and the result is output to the DVLA of the alarm detection (ALM) and the large deviation alarm (BB1) in the block memory.

Condition	Result
$DVL <  DV $	DVLA = BB1 = 1 <sup>*1</sup>
$(DVL - DVLS) <  DV  \leq DVL$	DVLA = BB1 = Last value status hold <sup>*1</sup>
$ DV  \leq (DVL - DVLS)$	DVLA = BB1 = 0

\*1 If the DMLI or ERRI of the disable alarm detection (INH) is set to 1, the DVLA of the alarm detection (ALM) and the large deviation alarm (BB1) are set to 0.

### ■Loop stop processing (6)

The following processing is performed according to the SPA status of the alarm detection (ALM).

SPA status	Processing details
1	The loop stops. When the loop stops, the following operations are performed and the S.2PID instruction ends. <ul style="list-style-type: none"> <li>• The output value (BW) is set to 0.</li> <li>• The DVLA of alarm detection (ALM) is set to 0.</li> <li>• The control mode (MODE) is set to MAN.</li> <li>• BB1 of BB is set to 0.</li> </ul>
0	The loop runs and "control cycle determination processing (7)" is performed.

### ■Control cycle determination (7)

If the specified control cycle is not reached, output value BW ( $\Delta MV$ ) is set to 0 and the S.2PID instruction is terminated. If the specified control cycle is reached, "SV setting processing (1)" is performed.

### Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1), (s2), or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.

# Position type PID control

## S.PIDP

Performs position type PID operation. The instruction performs the following processing steps: SV setting, tracking, gain (Kp) operation, PID operation, deviation check, and control mode determination. Depending on the operation result up to the mode determination processing, the instruction decides next processing: variation rate & upper/lower limiter and output conversion, or alarm clear and output conversion.

Ladder	ST
	ENO:=S_PIDP(EN,s1,s2,s3,d1,d2);

FBD/LD

### Execution condition

Instruction	Execution condition
S.PIDP	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
(s3)	When E2 is used: Set value start device When E2 is not used: Dummy device	Refer to "Set value".	Single-precision real number
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

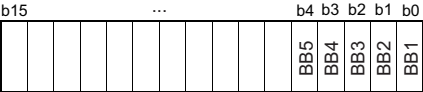
#### Applicable devices

Operand	Bit		Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□V□, J□□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	○	—	—	—	—	
(d2)	—	—	○	—	—	—	○	—	—	—	—	
(s3)	—	—	○	—	—	—	○	—	—	—	—	


## Input data

Operand: (s1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Input value	-999999 to 999999 [%]	Single-precision real number	—	User

## Block memory

Operand: (d1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999	Single-precision real number	—	System
+2	BB	BB1 Alarm	 <p>0: No alarm 1: Alarm</p>	16-bit unsigned binary	—	System
		BB2 Large deviation alarm				
		BB3 Output upper limit alarm				
		BB4 Output lower limit alarm				
		BB5 Output variation rate alarm				

## Operation constant

Operand: (s2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	MTD	Derivative gain	0 to 999999	Single-precision real number	8.0	User
+2 +3	DVLS	Large deviation alarm hysteresis	0 to 100 [%]	Single-precision real number	2.0	User
+4	PN	Operation mode	0: Reverse action 1: Direct action	16-bit unsigned binary	0	User
+5	TRK	Tracking bit	0: Tracking not performed 1: Tracking performed	16-bit unsigned binary	0	User
+6	SVPTN	Set value pattern	0 to 3  <p>(2) (1)</p> <p>(1) Use of set value Specify whether to use the set value (E2) or not. 0: Used 1: Not used (2) Set value pattern Specify whether to use the upper loop MV as the set value (E2) or not. 0: E2 is the upper loop MV. 1: E2 is not the upper loop MV.</p>	16-bit unsigned binary	3	User
+7 +8	NMAX	Output conversion upper limit	-999999 to 999999	Single-precision real number	100.0	User
+9 +10	NMIN	Output conversion lower limit	-999999 to 999999	Single-precision real number	0.0	User



Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+60 +61	GG	Gap gain	0 to 999999	Single-precision real number	1.0	User

**Loop tag past value memory**

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

The loop tag past value memory occupies 32 words after the loop tag memory.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+96	—	Control cycle counter initialization completion flag	—	—	—	System
+97	—	Control cycle counter (The value is rounded off to the nearest whole number.)				
+100 +101	I <sub>n-1</sub>	Last value				
+102 +103	B <sub>n-1</sub>	Last value				
+104 +105	PV <sub>n</sub>	Process value				
+106 +107	PV <sub>n-1</sub>	Last process value				
+116	ALM2	Alarm detection 2				

**Set value**

The set value (E2) is valid only when b0 of the set value pattern (SVPTN) is set to 0 (Used). To use the upper loop MV as the set value (E2), specify the device (offset +12) where the manipulated value (MV) of the upper loop is set.

If E2 is not used, specify a dummy device (SD820).

Operand: (s3)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E2	Set value	-10 to 110 [%]	Single-precision real number	0.0	User

**Execution cycle (ΔT)**

Set the execution cycle (ΔT) in SD816 and SD817 in single-precision real number.

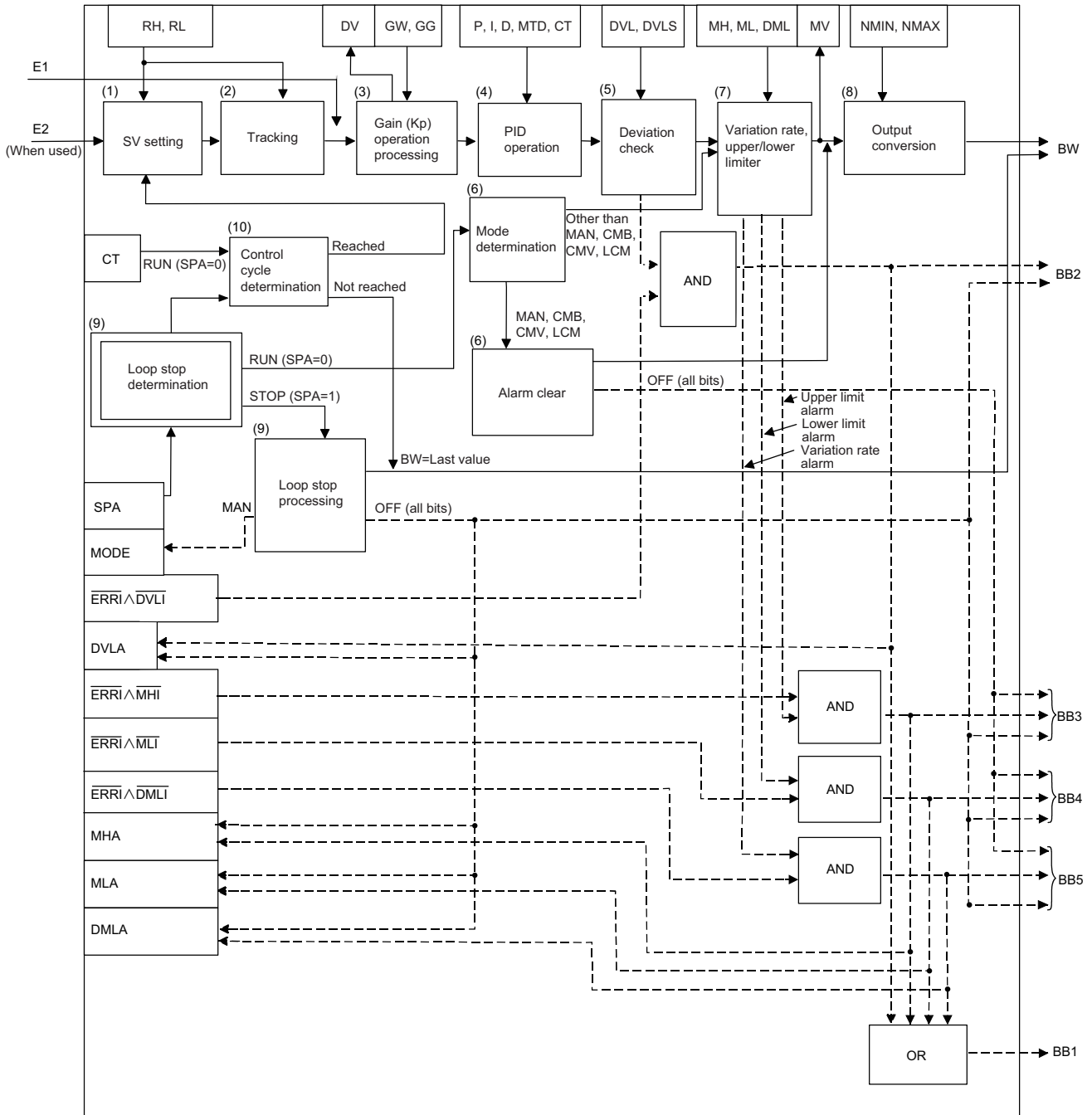
## Processing details

This instruction performs position type PID operation when the specified control cycle is reached.

At this time, the instruction also performs the following processing steps: SV setting, tracking, gain (Kp) operation, deviation check, and control mode (MODE) determination.

Depending on the operation result up to the mode determination processing, the instruction performs either variation rate & upper/lower limiter and output conversion processing, or alarm clear and output conversion processing.

The following is the processing block diagram of the S.PIDP instruction. (The numbers (1) to (10) in the diagram indicate the order of the processing.)



## ■SV setting processing (1)

The following processing is performed depending on the control mode (MODE) setting.

Control mode (MODE) setting	Processing details
CAS, CCB, CSV	<ul style="list-style-type: none"> <li>If the set value (E2) is specified, engineering value transformation processing (refer to the following expression) is performed, and then "tracking processing (2)" is performed.</li> </ul> $SV_n = \frac{RH-RL}{100} \times E2 + RL$ <ul style="list-style-type: none"> <li>If the set value (E2) is not specified, "tracking processing (2)" is performed without performing engineering value transformation processing.</li> </ul>
MAN, AUT, CMV, CMB, CAB, LCM, LCA, LCC	"Tracking processing (2)" is performed.

## ■Tracking processing (2)

- The set value (SV) is inversely transformed from the engineering value and  $SV_n'$  is calculated (refer to the following expression).

$$SV_n' = \frac{100}{RH-RL} \times (SV_n - RL)$$

- Tracking processing is performed when all of the following conditions are satisfied.
  - The tracking bit (TRK) is set to 1.
  - The set value (E2) is used.
  - The control mode (MODE) is set to any of the following: MAN, AUT, CMV, CMB, CAB, LCM, LCA, or LCC.  
E2 =  $SV_n'$
- If the set value (E2) is the upper loop MV, the TRKF of the disable alarm detection (INH) of the upper loop is set to 1.

## ■Gain (Kp) operation processing (3)

- The deviation (DV) is calculated under the following conditions.

Condition	Operational expression
Direct action (PN = 1)	$DV = E1 - SV_n'$
Reserve action (PN = 0)	$DV = SV_n' - E1$

- The output gain (K) is calculated under the following conditions.

Condition	Operational expression
$ DV  \leq GW$	$K = GG$
$ DV  > GW$	$K = 1 - \frac{(1-GG) \times GW}{ DV }$

## ■PID operation (4)

The PID operation is performed with the following operational expression.

Item		Operational expression
B <sub>n</sub>	Direct action (PN = 1)	$B_{n-1} + \frac{M_D \times T_D}{M_D \times CT + T_D} \times \left\{ (PV_n - PV_{n-1}) - \frac{CT \times B_{n-1}}{T_D} \right\}$
	Reserve action (PN = 0)	$B_{n-1} + \frac{M_D \times T_D}{M_D \times CT + T_D} \times \left\{ -(PV - PV_{n-1}) - \frac{CT \times B_{n-1}}{T_D} \right\}$
I <sub>n</sub>		$I_{n-1} + \frac{CT}{T_I} \times DV_n$
T		$K_p \times (DV_n + I_n + B_n)$

K<sub>p</sub>: K × Gain (P), M<sub>D</sub>: Derivative gain (MTD), T<sub>I</sub>: Integral constant (I), T<sub>D</sub>: Derivative constant (D)

Note that special processing is performed in the following cases.

Condition	Processing
In either of the following cases: 1. Derivative constant (D) = 0 (T <sub>D</sub> = 0) 2. Control mode (MODE) = MAN, LCM, or CMV	B <sub>n</sub> = 0 (Note that the loop tag past value memory is set.)
In any of the following cases: 1. Integral constant (I) = 0 (T <sub>I</sub> = 0) 2. MHA2 of the alarm detection 2 (ALM2) = 1 $\frac{CT}{T_I} \times DV_n > 0$ 3. MLA2 of the alarm detection 2 (ALM2) = 1 $\frac{CT}{T_I} \times DV_n < 0$ 4. Control mode (MODE) = MAN, LCM, or CMV	$\frac{CT}{T_I} \times DV_n = 0$
When all of the following conditions are satisfied: 1. b0 of SD818 (bumpless switching function of S.PIDP control) = 1. 2. TRKF of the disable alarm detection (INH) = 1 3. Control mode (MODE) = Other than MAN, LCM, and CMV	$I_{n-1} = \frac{MV}{K_p} - (DV_n + B_n)$ TRKF = 0

## ■Deviation check (5)

A deviation is checked under the following conditions, and the result is output to the DVLA of the alarm detection (ALM) and the large deviation alarm (BB2).

Condition	Result
DVL <  DV	DVLA = BB2 = 1*1
(DVL - DVLS) <  DV  ≤ DVL	DVLA = BB2 = Last value status hold*1
DV  ≤ (DVL - DVLS)	DVLA = BB2 = 0

\*1 If the DMLI or ERRI of the disable alarm detection (INH) is set to 1, the DVLA of the alarm detection (ALM) and the large deviation alarm (BB2) are set to 0.

## ■Mode determination (6)

The following processing is performed depending on the control mode (MODE).

Control mode (MODE)	Processing details
MAN, CMB, CMV, LCM (alarm clear processing)	<ul style="list-style-type: none"> <li>The MHA, MLA, and DMLA of the alarm detection (ALM) are set to 0.</li> <li>The MHA2 and MLA2 of the alarm detection 2 (ALM2) are set to 0.</li> <li>The alarm bits (BB3, BB4, and BB5) are set to 0.</li> <li>The data of BB2 is transferred to BB1. (BB1 = BB2)</li> <li>When b0 of SD818 (bumpless switching function of S.PIDP control) is 1, TRKF of INH is set to 1.</li> <li>"Output conversion processing (8)" is performed and the instruction ends.</li> </ul>
AUT, CAB, CAS, CCB, CSV, LCA, LCC	"Variation rate & upper/lower limiter processing (7)" is performed.



## ■Variation rate & upper/lower limiter (7)

The variation rate and upper/lower limits of the input value (E1) are checked, and the data after the processing and an alarm are output.

Variation rate limiter processing performs the following operations, and outputs the result to the output variation rate alarm (BB5) of (d1) and the DMLA of the alarm detection (ALM).

Condition	BB5, DMLA	T1
$ T - MV  \leq DML$	0	T
$(T - MV) > DML$	$1^{*1}$	$MV + DML$
$(T - MV) < -DML$	$1^{*1}$	$MV - DML$

\*1 If the DMLI or ERRI of the disable alarm detection (INH) is set to 1, the output variation rate alarm (BB5) and the DMLA of the alarm detection (ALM) are set to 0.

Upper/lower limiter processing performs the following operations, and outputs the result to the output upper limit alarm (BB3); output lower limit alarm (BB4); MHA and MLA of the alarm detection (ALM); and MHA2 and MLA2 of the alarm detection (ALM2).

Condition	BB4, MLA, MLA2 <sup>*4</sup>	BB3, MHA, MHA2 <sup>*4</sup>	MV
$T1 > MH$	0	$1^{*2}$	MH
$T1 < ML$	$1^{*3}$	0	ML
$ML \leq T1 \leq MH$	0	0	T1

\*2 If the MHI or ERRI of the disable alarm detection (INH) is set to 1, the output upper limit alarm (BB3) and the MHA of the alarm detection (ALM) are set to 0.

Note that the MHA2 of the alarm detection 2 (ALM2) remains 1.

\*3 If the MLI or ERRI of the disable alarm detection (INH) is set to 1, the output lower limit alarm (BB4) and the MLA of the alarm detection (ALM) are set to 0.

Note that the MLA2 of the alarm detection 2 (ALM2) remains 1.

\*4 If the specified control cycle is not reached, the MHA2 and MLA2 status of the alarm detection 2 (ALM2) are held.

## ■Output conversion processing (8)

The output value (BW) is calculated from the following expression.

$$BW = \frac{NMAX - NMIN}{100} \times MV + NMIN$$

## ■Loop stop processing (9)

The following processing is performed according to the SPA status of the alarm detection (ALM).

SPA status	Processing details
1	The loop stops. The following operations are performed and the instruction ends. <ul style="list-style-type: none"> <li>The last output value (BW) is held.</li> <li>The DVLA, MHA, MLA, and DMLA of the alarm detection (ALM) are set to 0.</li> <li>The MHA2 and MLA2 of the alarm detection 2 (ALM2) are set to 0.</li> <li>The control mode (MODE) is set to MAN.</li> <li>All of the alarm bits (BB1, BB2, BB3, BB4, and BB5) are set to 0.</li> </ul>
0	The loop runs and "control cycle determination processing (10)" is performed.

## ■Control cycle determination (10)

If the specified control cycle is not reached, "mode determination processing (6)" is performed regarding T as MV.

If the specified control cycle is reached, "SV setting processing (1)" is performed.

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1), (s2), or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.

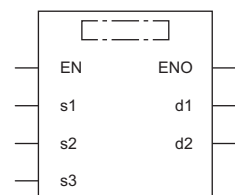
# Sample PI control

## S.SPI

The instruction checks whether ST or HT is applicable and, if ST is applicable, performs the following processing steps: SV setting, tracking, gain (Kp) operation, SPI operation, and deviation check.

Ladder	ST
	<pre>ENO:=S_SPI(EN,s1,s2,s3,d1,d2);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.SPI	

### Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
(s3)	When E2 is used: Set value start device When E2 is not used: Dummy device	Refer to "Set value".	Single-precision real number
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit


### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□\□, U3E□\□(H)G□	Z	LT, LST, LC	LZ		K	H	E	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	—	○	—	—	—	—	○	—	—	—	—


## Input data

Operand: (s1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Input value	-999999 to 999999 [%]	Single-precision real number	—	User

## Block memory

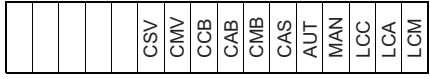
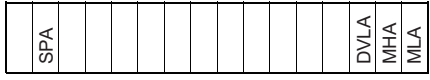
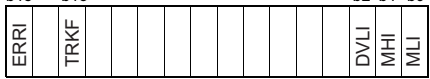
Operand: (d1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value ( $\Delta$ MV)	-999999 to 999999 [%]	Single-precision real number	—	System
+2	BB BB1	Large deviation alarm	 <p>0: No alarm 1: Alarm</p>	16-bit unsigned binary	—	System

## Operation constant

Operand: (s2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	DVLS	Large deviation alarm Hysteresis	0 to 100 [%]	Single-precision real number	2.0	User
+2	PN	Operation mode	0: Reverse action 1: Direct action	16-bit unsigned binary	0	User
+3	TRK	Tracking bit	0: Tracking not performed 1: Tracking performed	16-bit unsigned binary	0	User
+4	SVPTN	Set value pattern	 <p>(2) (1)</p> <p>(1) Use of set value Specify whether to use the set value (E2) or not. 0: Used 1: Not used (2) Set value pattern Specify whether to use the upper loop MV as the set value (E2) or not. 0: E2 is the upper loop MV. 1: E2 is not the upper loop MV.</p>	16-bit unsigned binary	3	User

## Loop tag memory

The loop tag memory occupies 96 words from the specified start device.

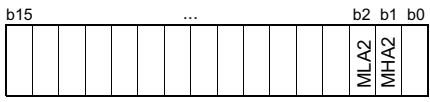
Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+1	MODE	Control mode	0 to FFFFH b15 ... b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 	16-bit unsigned binary	8H	User/system
+3	ALM	Alarm detection	0 to FFFFH b15 b14 ... b2 b1 b0  <p>SPA            0: Loop RUN            1: Loop STOP            DVLA, MHA, MLA            0: No alarm            1: Alarm</p>	16-bit unsigned binary	4000H	User/system
+4	INH	Disable alarm detection	0 to FFFFH b15 ... b13 ... b2 b1 b0  <p>TRKF            0: Tracking not set            1: Tracking set            ERRI, DVLI, MHI, MLI            0: Alarm detection enabled            1: Alarm detection disabled</p>	16-bit unsigned binary	0H	User/system
+14 +15	SV	Set value	RL to RH	Single-precision real number	0.0	User
+16 +17	DV	Deviation	-110 to 110 [%]	Single-precision real number	0.0	System
+22 +23	RH	Engineering value upper limit	-999999 to 999999	Single-precision real number	100.0	User
+24 +25	RL	Engineering value lower limit	-999999 to 999999	Single-precision real number	0.0	User
+46 +47	ST	Operating time	0 to 999999 [s] Set a value within the following range. $\frac{ST}{\Delta T} \leq 32767$	Single-precision real number	1.0	User
+50 +51	DVL	Deviation limit value	0 to 100 [%]	Single-precision real number	100.0	User
+52 +53	P	Gain	0 to 999999	Single-precision real number	1.0	User
+54 +55	I	Integral constant	0 to 999999 [s]	Single-precision real number	10.0	User
+56 +57	STHT	Sampling cycle	0 to 999999 [s] Set a value within the following range. $\frac{STHT}{\Delta T} \leq 32767$	Single-precision real number	0.0	User
+58 +59	GW	Gap width	0 to 100 [%]	Single-precision real number	0.0	User
+60 +61	GG	Gap gain	0 to 999999	Single-precision real number	1.0	User
+62 +63	MVP	MV internal operation value	-999999 to 999999 [%]	Single-precision real number	0.0	System

## Loop tag past value memory

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

The loop tag past value memory occupies 32 words after the loop tag memory.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+96	—	Control cycle counter initialization completion flag	—	—	—	System
+97		Sample counter (The value is rounded off to the nearest whole number.)				
+98		Operating counter (The value is rounded off to the nearest whole number.)				
+99		Hold counter				
+100 +101	DV <sub>n-1</sub>	Last deviation value				
+116	ALM2	Alarm detection 2	 <p>MHA2, MLA2 0: No alarm 1: Alarm</p>			

## Set value

The set value (E2) is valid only when b0 of the set value pattern (SVPTN) is set to 0 (Used). To use the upper loop MV as the set value (E2), specify the device (offset +12) where the manipulated value (MV) of the upper loop is set.

If E2 is not used, specify a dummy device (SD820).

Operand: (s3)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E2	Set value	-10 to 110 [%]	Single-precision real number	0.0	User

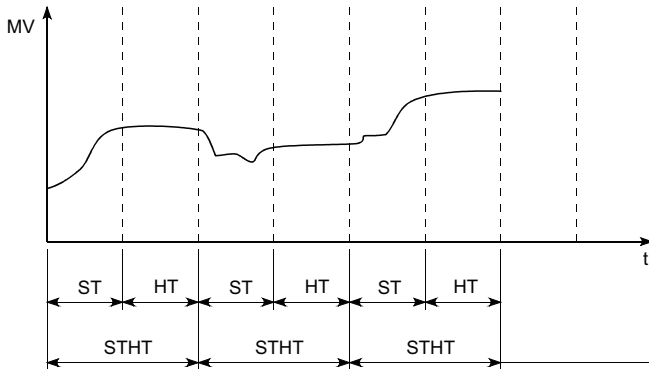
## Execution cycle (ΔT)

Set the execution cycle (ΔT) in SD816 and SD817 in single-precision real number.

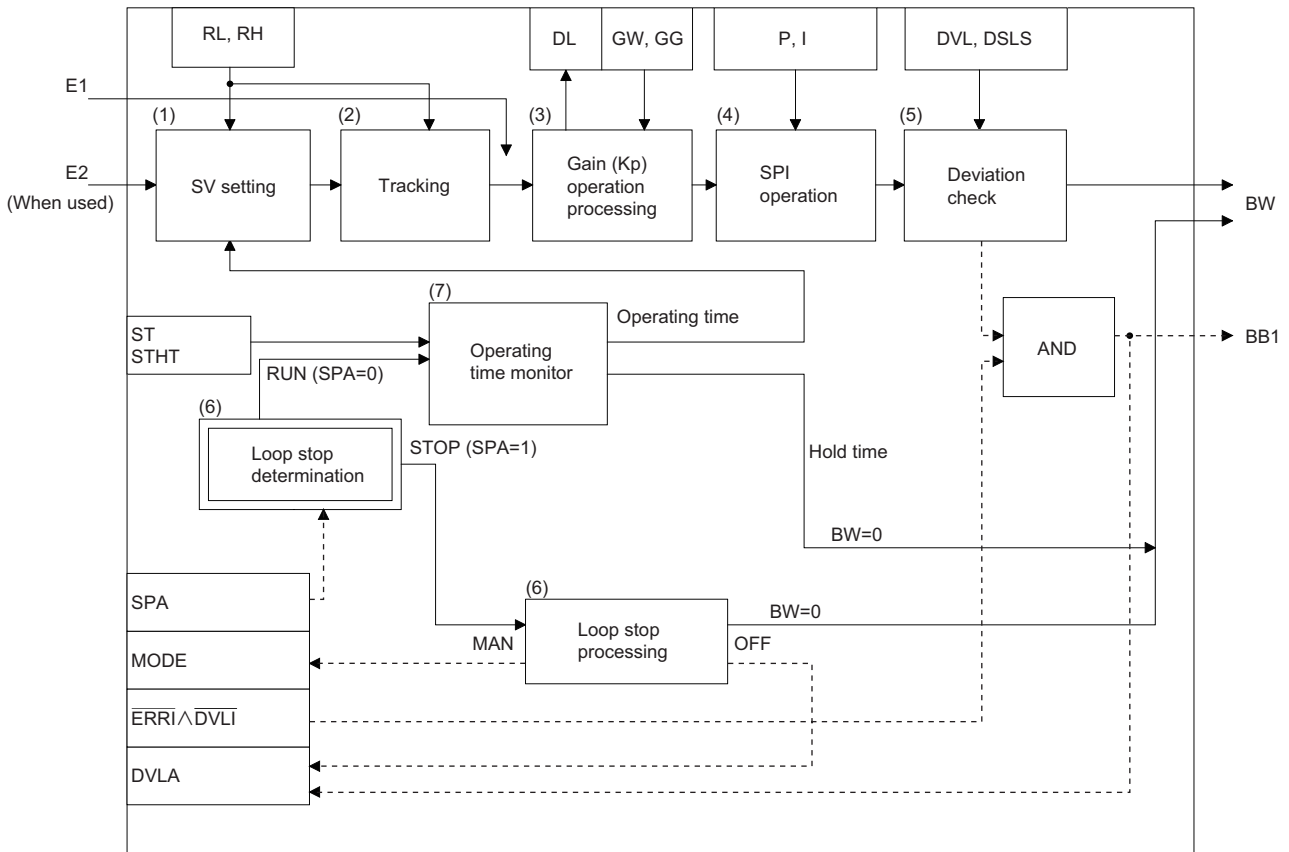
## Processing details

This instruction performs normal PI operation during the operating time (ST).

The instruction checks whether ST or HT is applicable and, if ST is applicable, performs the following processing steps: SV setting, tracking, gain ( $K_p$ ) operation, SPI operation, and deviation check.



The following is the processing block diagram of the S.SPI instruction. (The numbers (1) to (7) in the diagram indicate the order of the processing.)



## ■SV setting processing (1)

The following processing is performed depending on the control mode (MODE) setting.

Control mode (MODE)	Processing details
CAS, CCB, CSV	<ul style="list-style-type: none"> <li>If the set value (E2) is specified, engineering value transformation processing (refer to the following expression) is performed, and then "tracking processing (2)" is performed.</li> </ul> $SV_n = \frac{RH-RL}{100} \times E2 + RL$ <ul style="list-style-type: none"> <li>If the set value (E2) is not specified, "tracking processing (2)" is performed without performing engineering value transformation processing.</li> </ul>
MAN, AUT, CMV, CMB, CAB, LCM, LCA, LCC	"Tracking processing (2)" is performed.

## ■Tracking processing (2)

- The set value (SV) is inversely transformed from the engineering value and  $SV_n'$  is calculated (refer to the following expression).

$$SV_n' = \frac{100}{RH-RL} \times (SV_n - RL)$$

- Tracking processing is performed when all of the following conditions are satisfied.
  - The tracking bit (TRK) is set to 1.
  - The set value (E2) is used.
  - The control mode (MODE) is set to any of the following: MAN, AUT, CMV, CMB, CAB, LCM, LCA, or LCC.  
 $E2 = SV_n'$
- If the set value (E2) is the upper loop MV, the TRKF of the disable alarm detection (INH) of the upper loop is set to 1.

## ■Gain (Kp) operation processing (3)

- The deviation (DV) is calculated under the following conditions.

Condition	Operational expression
Direct action (PN = 1)	$DV = E1 - SV_n'$
Reserve action (PN = 0)	$DV = SV_n' - E1$

- The output gain (K) is calculated under the following conditions.

Condition	Operational expression
$ DV  \leq GW$	$K = GG$
$ DV  > GW$	$K = 1 - \frac{(1-GG) \times GW}{ DV }$

## ■SPI operation (4)

The SPI operation is performed with the following operational expression.

Item	Operational expression
During operating time (ST)	$BW = K_p \times \left\{ (DV_n - DV_{n-1}) + \frac{BT}{T_i} \times DV_n \right\}$
During hold time (sample cycle (STHT) - operating time (ST))	$BW = 0$ (Note that the loop tag past value memory is not set.)

$K_p$ : K×gain (P),  $T_i$ : integral constant (I), BT: execution cycle ( $\Delta T$ )

Note that special processing is performed in the following cases.

Condition	Processing
In any of the following cases: 1. Integral constant (I) = 0 (T <sub>I</sub> = 0) 2. MHA2 or MLA2 of alarm detection 2 (ALM2) is 1.  MVP > MH and $\frac{BT}{T_I} \times DV_n > 0$  3. MHA2 or MLA2 of alarm detection 2 (ALM2) is 1.  MVP < ML and $\frac{BT}{T_I} \times DV_n < 0$	$\frac{BT}{T_I} \times DV_n = 0$

### ■ Deviation check (5)

A deviation is checked under the following conditions, and the result is output to the DVLA of the alarm detection (ALM) and the large deviation alarm (BB1).

Condition	Result
DVL <  DV	DVLA = BB1 = 1*1
(DVL - DVLS) <  DV  ≤ DVL	DVLA = BB1 = Last value status hold*1
DV  ≤ (DVL - DVLS)	DVLA = BB1 = 0

\*1 If the DML1 or ERR1 of the disable alarm detection (INH) is set to 1, the DVLA of the alarm detection (ALM) and the large deviation alarm (BB1) are set to 0.

### ■ Loop stop processing (6)

The following processing is performed according to the SPA status of the alarm detection (ALM).

SPA status	Processing details
1	The loop stops. When the loop stops, the following operations are performed and the S.SPI instruction ends. <ul style="list-style-type: none"> <li>• The output value (BW) is set to 0.</li> <li>• The DVLA of alarm detection (ALM) is set to 0.</li> <li>• The control mode (MODE) is set to MAN.</li> <li>• BB1 of BB is set to 0.</li> </ul>
0	The loop runs and "operating time/hold time determination (7)" is performed.

### ■ Operating time monitoring (7)

The instruction checks whether ST or HT (= STHT - ST) is applicable and performs the following processing.

Condition	Processing details
Operating time (ST)	The instruction performs the following processing steps: SV setting, tracking, gain (Kp) operation, PI operation (operating time), and deviation check.
Hold time (HT) (= STHT - ST)	The instruction performs the following processing steps: tracking, SPI operation (hold time), and deviation check. Under the following conditions, however, PI control is performed continuously with the hold time set to 0.  $\frac{STHT}{\Delta T} \leq \frac{ST}{\Delta T}$ If the integral part of the left side of the above expression is 0, no processing is performed. (BW also remains unchanged.)



## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1), (s2), or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Operating time (ST) < 0
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	Sampling cycle (STHT) < 0
	(Operating time (ST) ÷ execution cycle ( $\Delta T$ )) > 32767
(Sample cycle (STHT) ÷ execution cycle ( $\Delta T$ )) > 32767	

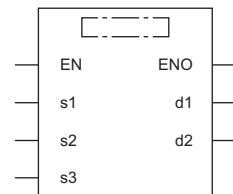
# I-PD control

## S.IPD

This instruction performs I-PD operation. The instruction performs the following processing steps: SV setting, tracking, gain  $K_P$  operation, IPD operation, and deviation check.

Ladder	ST
	<pre>ENO:=S_IPD(EN,s1,s2,s3,d1,d2);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.IPD	

### Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
(s3)	When E2 is used: Set value start device When E2 is not used: Dummy data	Refer to "Set value".	Single-precision real number
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

### Applicable devices


Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□\□, U3E□\□\G□	Z	LT, LST, LC	LZ		K	H	E	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	—	○	—	—	—	—	○	—	—	—	—

## Input data

Operand: (s1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Input value	-999999 to 999999 [%]	Single-precision real number	—	User

## Block memory


Operand: (d1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value ( $\Delta$ MV)	-999999 to 999999 [%]	Single-precision real number	—	System
+2	BB	BB1	Large deviation alarm	16-bit unsigned binary	—	System



0: No alarm  
1: Alarm

## Operation constant

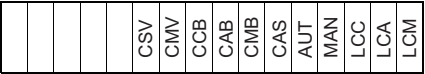
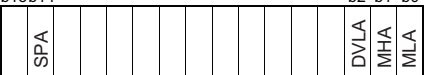
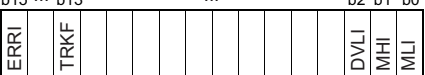
Operand: (s2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	MTD	Derivative gain	0 to 999999	Single-precision real number	8.0	User
+2 +3	DVLS	Large deviation alarm hysteresis	0 to 100 [%]	Single-precision real number	2.0	User
+4	PN	Operation mode	0: Reverse action 1: Direct action	16-bit unsigned binary	0	User
+5	TRK	Tracking bit	0: Tracking not performed 1: Tracking performed	16-bit unsigned binary	0	User
+6	SVPTN	Set value pattern	0 to 3	16-bit unsigned binary	3	User



(1) Use of set value  
Specify whether to use the set value (E2) or not.  
0: Used  
1: Not used  
(2) Set value pattern  
Specify whether to use the upper loop MV as the set value (E2) or not.  
0: E2 is the upper loop MV.  
1: E2 is not the upper loop MV.

## Loop tag memory

The loop tag memory occupies 96 words from the specified start device.

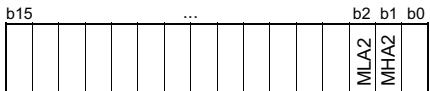
Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+1	MODE	Control mode	0 to FFFFH b15 ... b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 	16-bit unsigned binary	8H	User/system
+3	ALM	Alarm detection	0 to FFFFH b15 b14 ... b2 b1 b0  <p>SPA            0: Loop RUN            1: Loop STOP            DVLA, MHA, MLA            0: No alarm            1: Alarm</p>	16-bit unsigned binary	4000H	User/system
+4	INH	Disable alarm detection	0 to FFFFH b15 ... b13 ... b2 b1 b0  <p>TRKF            0: Tracking not set            1: Tracking set            ERRI, DVLI, MHI, MLI            0: Alarm detection enabled            1: Alarm detection disabled</p>	16-bit unsigned binary	0H	User/system
+14	SV	Set value	RL to RH	Single-precision real number	0.0	User
+15						
+16	DV	Deviation	-110 to 110 [%]	Single-precision real number	0.0	System
+17						
+22	RH	Engineering value upper limit	-999999 to 999999	Single-precision real number	100.0	User
+23						
+24	RL	Engineering value lower limit	-999999 to 999999	Single-precision real number	0.0	User
+25						
+46	CT	Control cycle	0 to 999999 [s] Set a value within the following range. $\frac{CT}{\Delta T} \leq 32767$	Single-precision real number	1.0	User
+47						
+50	DVL	Deviation limit value	0 to 100 [%]	Single-precision real number	100.0	User
+51						
+52	P	Gain	0 to 999999	Single-precision real number	1.0	User
+53						
+54	I	Integral constant	0 to 999999 [s]	Single-precision real number	10.0	User
+55						
+56	D	Derivative constant	0 to 999999 [s]	Single-precision real number	0.0	User
+57						
+58	GW	Gap width	0 to 100 [%]	Single-precision real number	0.0	User
+59						
+60	GG	Gap gain	0 to 999999	Single-precision real number	1.0	User
+61						
+62	MVP	MV internal operation value	-999999 to 999999 [%]	Single-precision real number	0.0	System
+63						

## Loop tag past value memory

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

The loop tag past value memory occupies 32 words after the loop tag memory.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+96	—	Control cycle counter initialization completion flag	—	—	—	System
+97		Control cycle counter (The value is rounded off to the nearest whole number.)				
+102	$B_{n-1}$	Last value				
+103						
+104	$PV_n$	Process value				
+105						
+106	$PV_{n-1}$	Last process value				
+107						
+108	$PV_{n-2}$	Last-but-one process value				
+109						
+116	ALM2	Alarm detection 2	 <p>MHA2, MLA2 0: No alarm 1: Alarm</p>			

## Set value

The set value (E2) is valid only when b0 of the set value pattern (SVPTN) is set to 0 (Used). To use the upper loop MV as the set value (E2), specify the device (offset +12) where the manipulated value (MV) of the upper loop is set.

If E2 is not used, specify a dummy device (SD820).

Operand: (s3)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E2	Set value	-10 to 110 [%]	Single-precision real number	0.0	User
+1						

## Execution cycle ( $\Delta T$ )

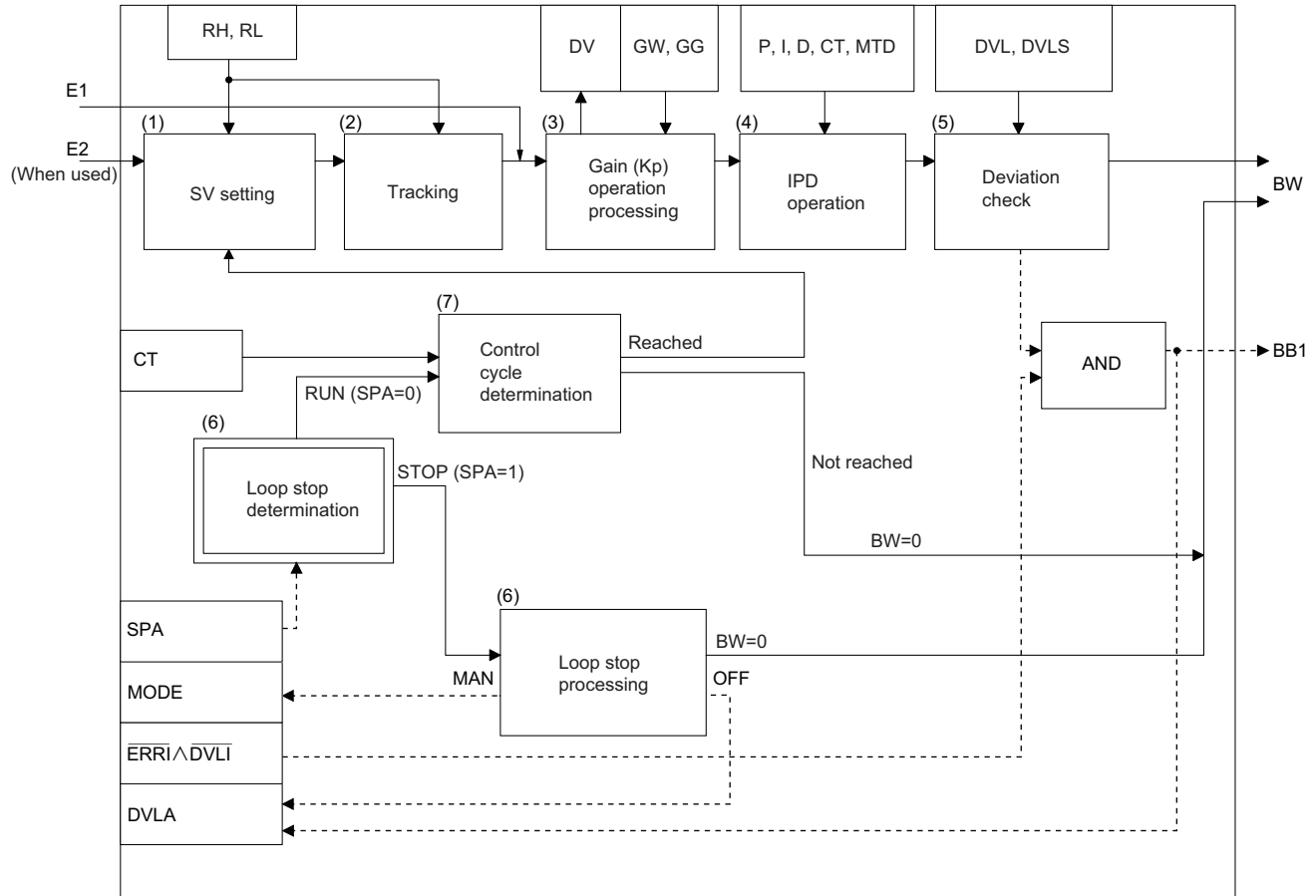
Set the execution cycle ( $\Delta T$ ) in SD816 and SD817 in single-precision real number.

## Processing details

This instruction performs I-PD control when the specified control cycle is reached.

At this time, the instruction also performs the following processing steps: SV setting, tracking, gain (Kp) operation, and deviation check.

The following is the processing block diagram of the S.IPD instruction. (The numbers (1) to (7) in the diagram indicate the order of the processing.)



### SV setting processing (1)

The following processing is performed depending on the control mode (MODE) setting.

Control mode (MODE) setting	Processing details
CAS, CCB, CSV	<ul style="list-style-type: none"> <li>If the set value (E2) is specified, engineering value transformation processing (refer to the following expression) is performed, and then "tracking processing (2)" is performed.</li> </ul> $SV_n = \frac{RH-RL}{100} \times E2 + RL$ <ul style="list-style-type: none"> <li>If the set value (E2) is not specified, "tracking processing (2)" is performed without performing engineering value transformation processing.</li> </ul>
MAN, AUT, CMV, CMB, CAB, LCM, LCA, LCC	"Tracking processing (2)" is performed.

### Tracking processing (2)

- The set value (SV) is inversely transformed from the engineering value and  $SV_n'$  is calculated (refer to the following expression).

$$SV_n' = \frac{100}{RH-RL} \times (SV_n - RL)$$

- Tracking processing is performed when all of the following conditions are satisfied.
  - The tracking bit (TRK) is set to 1.
  - The set value (E2) is used.
  - The control mode (MODE) is set to any of the following: MAN, AUT, CMV, CMB, CAB, LCM, LCA, or LCC.  $E2 = SV_n'$
- If the set value (E2) is the upper loop MV, the TRKF of the disable alarm detection (INH) of the upper loop is set to 1.

### ■ Gain (Kp) operation processing (3)

- The deviation (DV) is calculated under the following conditions.

Condition	Operational expression
Direct action (PN = 1)	$DV = E1 - SV_n'$
Reserve action (PN = 0)	$DV = SV_n' - E1$

- The output gain (K) is calculated under the following conditions.

Condition	Operational expression
$ DV  \leq GW$	$K = GG$
$ DV  > GW$	$K = 1 - \frac{(1-GG) \times GW}{ DV }$

### ■ I-PD operation (4)

The PID operation is performed with the following operational expression.

Item	Operational expression
B <sub>n</sub>	Direct action (PN = 1) $B_{n-1} + \frac{M_D \times T_D}{M_D \times CT + T_D} \times \left\{ (PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times B_{n-1}}{T_D} \right\}$
	Reserve action (PN = 0) $B_{n-1} + \frac{M_D \times T_D}{M_D \times CT + T_D} \times \left\{ -(PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times B_{n-1}}{T_D} \right\}$
BW(ΔMV)	Direct action (PN = 1) $K_P \times \left\{ \frac{CT}{T_I} \times DV_n + (PV_n - PV_{n-1}) + B_n \right\}$
	Reserve action (PN = 0) $K_P + \left\{ \frac{CT}{T_I} \times DV_n - (PV_n - PV_{n-1}) + B_n \right\}$

K<sub>P</sub>: K × Gain (P), M<sub>D</sub>: Derivative gain (MTD), T<sub>I</sub>: Integral constant (I), T<sub>D</sub>: Derivative constant (D)

Note that special processing is performed in the following cases.

Condition	Processing
In either of the following cases: 1. Derivative constant (D) = 0 (T <sub>D</sub> = 0) 2. Control mode (MODE) = MAN, LCM, or CMV	B <sub>n</sub> = 0 (Note that the loop tag past value memory is set.)
In any of the following cases: 1. Integral constant (I) = 0 (T <sub>I</sub> = 0) 2. MHA2 or MLA2 of alarm detection 2 (ALM2) is 1.  MVP > MH  and $\frac{CT}{T_I} \times DV_n > 0$  3. MHA2 or MLA2 of alarm detection 2 (ALM2) is 1.  MVP < ML  and $\frac{CT}{T_I} \times DV_n < 0$	$\frac{CT}{T_I} \times DV_n = 0$

### ■ Deviation check (5)

A deviation is checked under the following conditions, and the result is output to the DVLA of the alarm detection (ALM) and the large deviation alarm (BB1).

Condition	Result
$DVL <  DV $	DVLA = BB1 = 1*1
$(DVL - DVLS) <  DV  \leq DVL$	DVLA = BB1 = Last value status hold*1
$ DV  \leq (DVL - DVLS)$	DVLA = BB1 = 0

\*1 If the DMLI or ERRI of the disable alarm detection (INH) is set to 1, the DVLA of the alarm detection (ALM) and the large deviation alarm (BB1) are set to 0.

## ■ Loop stop processing (6)

The following processing is performed according to the SPA status of the alarm detection (ALM).

SPA status	Processing details
1	The loop stops. When the loop stops, the following operations are performed and the S.IPD instruction ends. <ul style="list-style-type: none"><li>• The output value (BW) is set to 0.</li><li>• The DVLA of alarm detection (ALM) is set to 0.</li><li>• The control mode (MODE) is set to MAN.</li><li>• BB1 of BB is set to 0.</li></ul>
0	The loop runs and "control cycle determination processing (7)" is performed.

## ■ Control cycle determination (7)

If the specified control cycle is not reached, output value (BW) is set to 0 and the S.IPD instruction is terminated.

If the specified control cycle is reached, "SV setting processing (1)" is performed.

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1), (s2), or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.



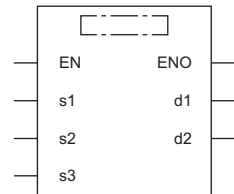
# Blend PI control

## S.BPI

This instruction performs blend PI operation. The instruction performs the following processing steps: SV setting, tracking, gain  $K_P$  operation, BPI operation, and deviation check.

Ladder	ST
	<pre>ENO:=S_BPI(EN,s1,s2,s3,d1,d2);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.BPI	

### Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
(s3)	When E2 is used: Set value start device When E2 is not used: Dummy device	Refer to "Set value".	Single-precision real number
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

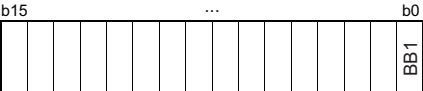
### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□, U3E□\(\H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	—	○	—	—	—	—	○	—	—	—	—

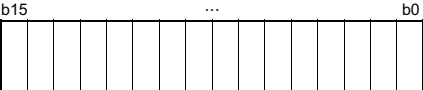
## Input data

Operand: (s1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Input value	-999999 to 999999 [%]	Single-precision real number	—	User

## Block memory

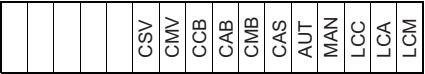
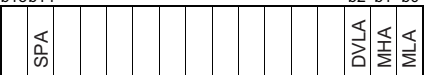
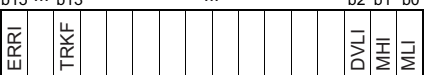
Operand: (d1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value ( $\Delta$ MV)	-999999 to 999999 [%]	Single-precision real number	—	System
+2	BB BB1	Large deviation alarm	 <p>0: No alarm 1: Alarm</p>	16-bit unsigned binary	—	System

## Operation constant

Operand: (s2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	DVLS	Large deviation alarm hysteresis	0 to 100 [%]	Single-precision real number	2.0	User
+2	PN	Operation mode	0: Reverse action 1: Direct action	16-bit unsigned binary	0	User
+3	TRK	Tracking bit	0: Tracking not performed 1: Tracking performed	16-bit unsigned binary	0	User
+4	SVPTN	Set value pattern	 <p>(2) (1)</p> <p>(1) Use of set value Specify whether to use the set value (E2) or not. 0: Used 1: Not used (2) Set value pattern Specify whether to use the upper loop MV as the set value (E2) or not. 0: E2 is the upper loop MV. 1: E2 is not the upper loop MV.</p>	16-bit unsigned binary	3	User

## Loop tag memory

The loop tag memory occupies 96 words from the specified start device.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+1	MODE	Control mode	0 to FFFFH b15 ... b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 	16-bit unsigned binary	8H	User/system
+3	ALM	Alarm detection	0 to FFFFH b15 b14 ... b2 b1 b0  <p>SPA            0: Loop RUN            1: Loop STOP            DVLA, MHA, MLA            0: No alarm            1: Alarm</p>	16-bit unsigned binary	4000H	User/system
+4	INH	Disable alarm detection	0 to FFFFH b15 ... b13 ... b2 b1 b0  <p>TRKF            0: Tracking not set            1: Tracking set            ERRI, DVLI, MHI, MLI            0: Alarm detection enabled            1: Alarm detection disabled</p>	16-bit unsigned binary	0H	User/system
+14 +15	SV	Set value	RL to RH	Single-precision real number	0.0	User
+16 +17	DV	Deviation	-110 to 110 [%]	Single-precision real number	0.0	System
+22 +23	RH	Engineering value upper limit	-999999 to 999999	Single-precision real number	100.0	User
+24 +25	RL	Engineering value lower limit	-999999 to 999999	Single-precision real number	0.0	User
+46 +47	CT	Control cycle	0 to 999999 [s] Set a value within the following range. $\frac{CT}{\Delta T} \leq 32767$	Single-precision real number	1.0	User
+50 +51	DVL	Deviation limit value	0 to 100 [%]	Single-precision real number	100.0	User
+52 +53	P	Gain	0 to 999999	Single-precision real number	1.0	User
+54 +55	I	Integral constant	0 to 999999 [s]	Single-precision real number	10.0	User
+56 +57	SDV	DV cumulative total ( $\Sigma DV$ )	-999999 to 999999 [%]	Single-precision real number	0.0	System
+58 +59	GW	Gap width	0 to 100 [%]	Single-precision real number	0.0	User
+60 +61	GG	Gap gain	0 to 999999	Single-precision real number	1.0	User

### ■Loop tag past value memory

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

The loop tag past value memory occupies 32 words after the loop tag memory.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+96	—	Control cycle counter initialization completion flag	—	—	—	System
+97		Control cycle counter (The value is rounded off to the nearest whole number.)				
+98 +99		—				

### ■Set value

The set value (E2) is valid only when b0 of the set value pattern (SVPTN) is set to 0 (Used). To use the upper loop MV as the set value (E2), specify the device (offset +12) where the manipulated value (MV) of the upper loop is set.

If E2 is not used, specify a dummy device (SD820).

Operand: (s3)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E2	Set value	-10 to 110 [%]	Single-precision real number	0.0	User

### ■Execution cycle ( $\Delta T$ )

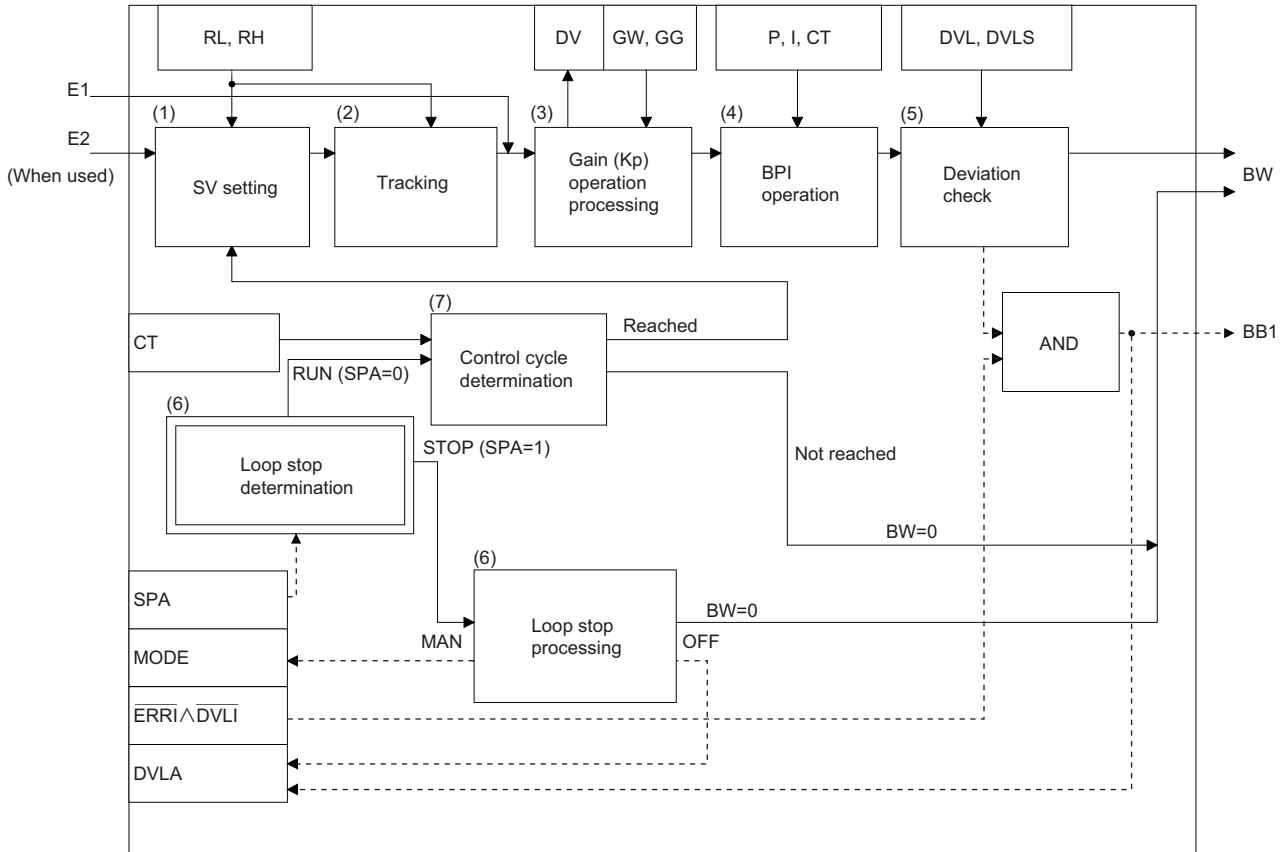
Set the execution cycle ( $\Delta T$ ) in SD816 and SD817 in single-precision real number.

## Processing details

This instruction performs BPI operation when the specified control cycle is reached.

At this time, the instruction also performs the following processing steps: SV setting, tracking, gain (Kp) operation, and deviation check.

The following is the processing block diagram of the S.BPI instruction. (The numbers (1) to (7) in the diagram indicate the order of the processing.)



### SV setting processing (1)

The following processing is performed depending on the control mode (MODE) setting.

Control mode (MODE) setting	Processing details
CAS, CCB, CSV	<ul style="list-style-type: none"> <li>If the set value (E2) is specified, engineering value transformation processing (refer to the following expression) is performed, and then "tracking processing (2)" is performed.</li> </ul> $SV_n = \frac{RH-RL}{100} \times E2 + RL$ <ul style="list-style-type: none"> <li>If the set value (E2) is not specified, "tracking processing (2)" is performed without performing engineering value transformation processing.</li> </ul>
MAN, AUT, CMV, CMB, CAB, LCM, LCA, LCC	"Tracking processing (2)" is performed.

### Tracking processing (2)

- The set value (SV) is inversely transformed from the engineering value and  $SV_n'$  is calculated (refer to the following expression).

$$SV_n' = \frac{100}{RH-RL} \times (SV_n - RL)$$

- Tracking processing is performed when all of the following conditions are satisfied.
  - The tracking bit (TRK) is set to 1.
  - The set value (E2) is used.
  - The control mode (MODE) is set to any of the following: MAN, AUT, CMV, CMB, CAB, LCM, LCA, or LCC.  $E2 = SV_n'$
- If the set value (E2) is the upper loop MV, the TRKF of the disable alarm detection (INH) of the upper loop is set to 1.

### ■Gain (Kp) operation processing (3)

- The deviation (DV) is calculated under the following conditions.

Condition	Operational expression
Direct action (PN = 1)	$DV = E1 - SV_n'$
Reserve action (PN = 0)	$DV = SV_n' - E1$

- The output gain (K) is calculated under the following conditions.

Condition	Operational expression
$ DV  \leq GW$	$K = GG$
$ DV  > GW$	$K = 1 - \frac{(1-GG) \times GW}{ DV }$

### ■BPI operation (4)

The BPI operation is performed with the following operational expression.

Item	Operational expression
BW(ΔMV)	$K_p \times BT \times (DV_n + \frac{CT}{T_i} \times \Sigma DV_i)$

K<sub>p</sub>: K×gain (P), BT: execution cycle, T<sub>i</sub>: integral constant (I), ΣDV<sub>i</sub>: DV<sub>n</sub> cumulative value, DV<sub>n</sub>: deviation

Note that special processing is performed in the following cases.

Condition	Processing
In either of the following cases: 1. Integral constant (I) = 0 (T <sub>i</sub> = 0) 2. MHA or MLA of alarm detection (ALM) is 1.	$\frac{CT}{T_i} \times \Sigma DV_i = \text{Last value}$
Integral constant (I) ≠ 0 (T <sub>i</sub> ≠ 0)	$\frac{CT}{T_i} \times \Sigma DV_i = \frac{CT}{T_i} \times (\Sigma DV_i + DV_n)$

### ■Deviation check (5)

A deviation is checked under the following conditions, and the result is output to the DVLA of the alarm detection (ALM) and the large deviation alarm (BB1) in the device specified by (d2).

Condition	Result
$DVL <  DV $	$DVLA = BB1 = 1^{*1}$
$(DVL - DVLS) <  DV  \leq DVL$	$DVLA = BB1 = \text{Last value status hold}^{*1}$
$ DV  \leq (DVL - DVLS)$	$DVLA = BB1 = 0$

\*1 If the DMLI or ERRI of the disable alarm detection (INH) is set to 1, the DVLA of the alarm detection (ALM) and the large deviation alarm (BB1) are set to 0.

### ■Loop stop processing (6)

The following processing is performed according to the SPA status of the alarm detection (ALM).

SPA status	Processing details
1	The loop stops. When the loop stops, the following operations are performed and the S.BPI instruction ends. <ul style="list-style-type: none"> <li>The output value (BW) is set to 0.</li> <li>The DVLA of alarm detection (ALM) is set to 0.</li> <li>The control mode (MODE) is set to MAN.</li> <li>BB1 of BB is set to 0.</li> </ul>
0	The loop runs and "control cycle determination processing (7)" is performed.

### ■Control cycle determination (7)

If the specified control cycle is not reached, output value (BW) is set to 0 and the S.BPI instruction is terminated.

If the specified control cycle is reached, "SV setting processing (1)" is performed.

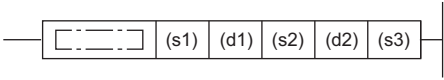
## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1), (s2), or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.

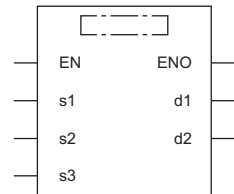
# Ratio calculation

## S.R


This instruction performs the following steps for the input data: engineering value transformation, tracking, variation rate limiter, and ratio calculation.

Ladder	ST
	<pre>ENO:=S_R(EN,s1,s2,s3,d1,d2);</pre>

## FBD/LD



## Execution condition

Instruction	Execution condition
S.R	

## Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Single-precision real number
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
(s3)	When E2 is used: Set value start device When E2 is not used: Dummy device	Refer to "Set value".	Single-precision real number
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

### Applicable devices

Operand	Bit		Word				Double word		Indirect specification	Constant				Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□\□, U3E□(H)G□	Z	LT, LST, LC	LZ	K, H		E	\$			
(s1)	—	—	○	—	—	—	○	—	—	—	—	—	—	
(d1)	—	—	○	—	—	—	○	—	—	—	—	—	—	
(s2)	—	—	○	—	—	—	○	—	—	—	—	—	—	
(d2)	—	—	○	—	—	—	○	—	—	—	—	—	—	
(s3)	—	—	○	—	—	—	○	—	—	—	—	—	—	



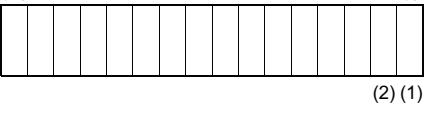
## Input data

Operand: (s1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Input value	-999999 to 999999 [%]	Single-precision real number	—	User

## Block memory

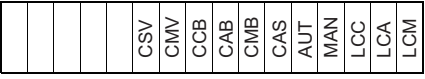

Operand: (d1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999 [%]	Single-precision real number	—	System

## Operation constant

Operand: (s2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	TRK	Tracking bit	0: Tracking not performed 1: Tracking performed	16-bit unsigned binary	0	User
+1	SVPTN	Set value pattern	0 to 3  <p>(1) Use of set value Specify whether to use the set value (E2) or not. 0: Used 1: Not used (2) Set value pattern Specify whether to use the upper loop MV as the set value (E2) or not. 0: E2 is the upper loop MV. 1: E2 is not the upper loop MV.</p>	16-bit unsigned binary	3	User

## Loop tag memory

The loop tag memory occupies 96 words from the specified start device.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+1	MODE	Control mode	0 to FFFFH <div style="display: flex; justify-content: space-between; align-items: center;"> <span>b15</span> <span>...</span> <span>b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0</span> </div> 	16-bit unsigned binary	8H	User/system
+3	ALM	Alarm detection	0 to FFFFH <div style="display: flex; justify-content: space-between; align-items: center;"> <span>b15 b14</span> <span>...</span> <span>b0</span> </div>  <p>SPA            0: Loop RUN            1: Loop STOP</p>	16-bit unsigned binary	4000H	User/system
+14 +15	SPR	Set value	-999999 to 999999	Single-precision real number	0.0	User
+16 +17	BIAS	Bias	-999999 to 999999 [%]	Single-precision real number	0.0	User
+46 +47	CT	Control cycle	0 to 999999 [s] Set a value within the following range. $\frac{CT}{\Delta T} \leq 32767$	Single-precision real number	1.0	User
+50 +51	DR	Variation rate limit value	0 to 999999	Single-precision real number	100.0	User
+52 +53	RMAX	Ratio upper limit value	-999999 to 999999	Single-precision real number	100.0	User
+54 +55	RMIN	Ratio lower limit value	-999999 to 999999	Single-precision real number	0.0	User
+56 +57	R <sub>n</sub>	Ratio current value	-999999 to 999999	Single-precision real number	0.0	System

## Loop tag past value memory

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

The loop tag past value memory occupies 32 words after the loop tag memory.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+96	—	Control cycle counter initialization completion flag	—	—	—	System
+97	—	Control cycle counter (The value is rounded off to the nearest whole number.)	—	—	—	System
+98 +99	R <sub>n-1</sub>	Last value	—	—	—	System

### ■Set value

The set value (E2) is valid only when b0 of the set value pattern (SVPTN) is set to 0 (Used). To use the upper loop MV as the set value (E2), specify the device (offset +12) where the manipulated value (MV) of the upper loop is set.

If E2 is not used, specify a dummy device (SD820).

Operand: (s3)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E2	Set value	-10 to 110 [%]	Single-precision real number	0.0	User
+1						

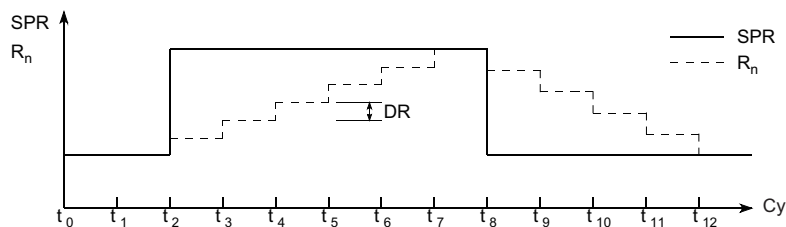
### ■Execution cycle ( $\Delta T$ )

Set the execution cycle ( $\Delta T$ ) in SD816 and SD817 in single-precision real number.

### Processing details

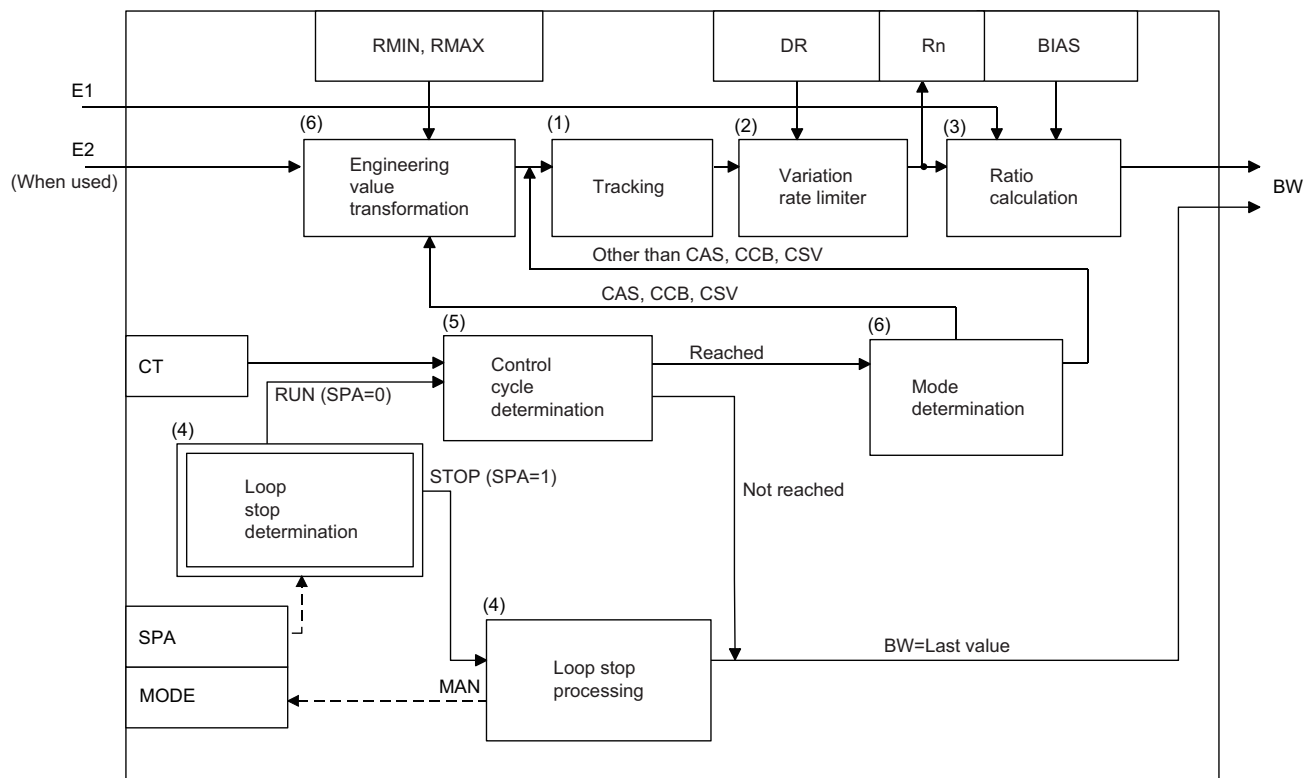
This instruction performs ratio calculation when the specified control cycle is reached.

At this time, the instruction also performs the following steps: control mode (MODE) determination, engineering value transformation, tracking, and variation rate limiter.



SPR: Set value,  $R_n$ : Current ratio value, DR: Variation rate limit value, Cy: Control cycle

The following is the processing block diagram of the S.R instruction. (The numbers (1) to (6) in the diagram indicate the order of the processing.)



## ■Tracking processing (1)

- Tracking processing is performed when all of the following conditions are satisfied.
- The tracking bit (TRK) is set to 1.
- The set value (E2) is used.
- The control mode (MODE) is set to any of the following: MAN, AUT, CMV, CMB, CAB, LCM, LCA, or LCC.

$$E2 = \frac{100}{R_{MAX} - R_{MIN}} \times (SPR - R_{MIN})$$

- If the set value (E2) is the upper loop MV, the TRKF of the disable alarm detection (INH) of the upper loop is set to 1.

## ■Variation rate limiter (2)

Variation rate limiter processing performs the following operations, and stores the result in the current ratio value ( $R_n$ ).

Condition	Operational expression
$(SPR - R_n) \geq DR$	$R_n = R_{n-1} + DR$
$(SPR - R_n) \leq -DR$	$R_n = R_{n-1} - DR$
$ SPR - R_n  < DR$	$R_n = SPR$

## ■Ratio calculation (3)

The ratio calculation is performed with the following operational expression.

$$BW = \frac{R_n - R_{MIN}}{R_{MAX} - R_{MIN}} \times E1 + BIAS$$

## ■Loop stop processing (4)

The following processing is performed according to the SPA status of the alarm detection (ALM).

SPA status	Processing details
1	The loop stops. When the loop stops, the following operations are performed and the S.R instruction ends. <ul style="list-style-type: none"> <li>• The last output value (BW) is held.</li> <li>• The control mode (MODE) is set to MAN.</li> </ul>
0	The loop runs and "control cycle determination processing (5)" is performed.

## ■Control cycle determination (5)

If the specified control cycle is not reached, output value (BW) is set to 0 and the S.R instruction is terminated.

If the specified control cycle is reached, "mode determination (6)" is performed.

## ■Mode determination (6)

The following processing is performed depending on the control mode (MODE).

Control mode (MODE)	Processing details
CAS, CCB, CSV	<ul style="list-style-type: none"> <li>• If the set value (E2) is specified, engineering value transformation processing (refer to the following expression) is performed, and then "variation rate limiter (2)" is performed.</li> </ul> $SPR = \frac{R_{MAX} - R_{MIN}}{100} \times E2 + R_{MIN}$ <ul style="list-style-type: none"> <li>• If the set value (E2) is not specified, "variation rate limiter (2)" is performed without performing engineering value transformation processing.</li> </ul>
MAN, AUT, CMV, CMB, CAB, LCM, LCA, LCC	"Tracking processing (1)" is performed.

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1) or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.

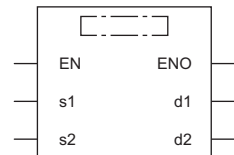
# Upper/lower limit alarm

## S.PHPL

This instruction checks whether the input data exceeds the upper limit or underruns the lower limit.

Ladder	ST
	<pre>ENO:=S_PHPL(EN,s1,s2,d1,d2);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.PHPL	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

#### Applicable devices

Operand	Bit		Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□□G□, J□□□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H, E, \$			
(s1)	—	—	○	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	○	—	—	—	—	
(s2)	—	—	—	—	—	—	—	—	—	○	—	
(d2)	—	—	○	—	—	—	○	—	—	—	—	

#### Input data

Operand: (s1)							
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by	
+0	E1	Input value	-999999 to 999999 [%]	Single-precision real number	—	User	
+1							



Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+40 +41	HS	Upper/lower limit alarm hysteresis	0 to 999999 [%]	Single-precision real number	0.0	User
+42 +43	CTIM	Variation rate alarm check time	0 to 999999 [s] Set a value within the following range. $\frac{CTIM}{\Delta T} \leq 32767$	Single-precision real number	0.0	User
+44 +45	DPL	Variation rate alarm value	0 to 100 [%]	Single-precision real number	100.0	User

**Loop tag past value memory**

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

The loop tag past value memory occupies 32 words after the loop tag memory.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+124	—	Variation rate monitoring counter initialization completion flag	—	—	—	System
+125	—	Variation rate monitoring counter (rounded off to the nearest whole number)	—	—	—	System
+126 +127	E1 <sub>n-m</sub>	—	—	—	—	System

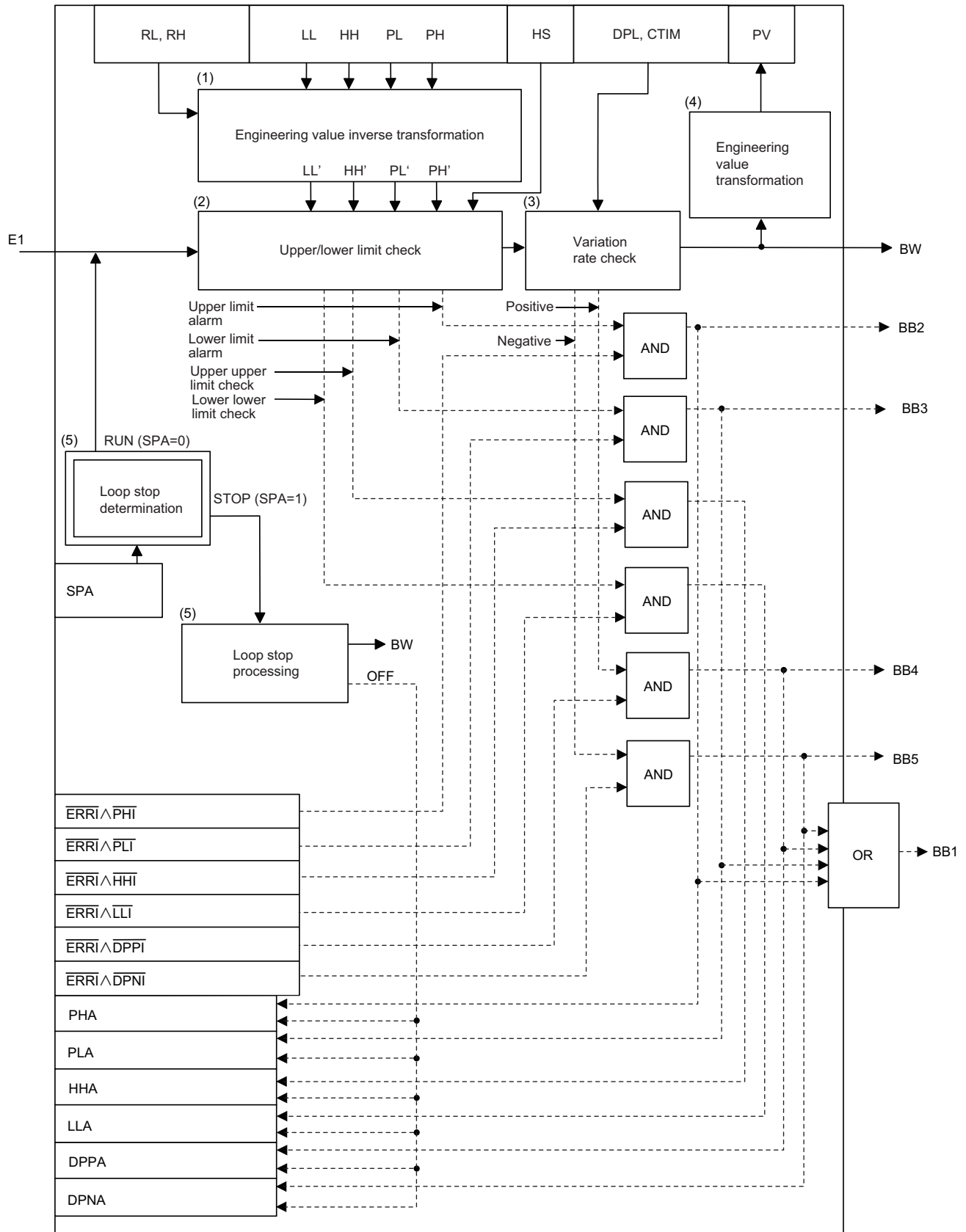
**Execution cycle (ΔT)**

Set the execution cycle (ΔT) in SD816 and SD817 in single-precision real number.

## Processing details

This instruction checks whether the input value (E1) exceeds the upper limit or underruns the lower limit, and outputs an alarm based on the result.

The following is the processing block diagram of the S.PHPL instruction. (The numbers (1) to (5) in the diagram indicate the order of the processing.)





### Engineering value inverse transformation (1)

The instruction performs the following operations to match the ranges of PH, PL, HH, and LL to the input values (E1).

$$PH' = \frac{100}{RH-RL} \times (PH-RL)$$

$$PL' = \frac{100}{RH-RL} \times (PL-RL)$$

$$HH' = \frac{100}{RH-RL} \times (HH-RL)$$

$$LL' = \frac{100}{RH-RL} \times (LL-RL)$$

### Upper/lower limit check (2)

The upper and lower limits of the input value (E1) are checked under the following conditions.

Check item	Condition	ALM	BB2	BB3
Upper limit check	E1 > PH'	PHA = 1 <sup>*1</sup>	1 <sup>*1</sup>	—
	E1 ≤ PH' - HS	PHA = 0	0	—
	Others	PHA: The last value is held. <sup>*1</sup>	Hold <sup>*1</sup>	—
Lower limit check	E1 < PL'	PLA = 1 <sup>*1</sup>	—	1 <sup>*1</sup>
	E1 ≥ PL' + HS	PLA = 0	—	0
	Others	PLA: The last value is held. <sup>*1</sup>	—	Hold <sup>*1</sup>
Upper upper limit check	E1 > HH'	HHA = 1 <sup>*1</sup>	—	—
	E1 ≤ HH' - HS	HHA = 0	—	—
	Others	HHA: The last value is held. <sup>*1</sup>	—	—
Lower lower limit check	E1 < LL'	LLA = 1 <sup>*1</sup>	—	—
	E1 ≥ LL' + HS	LLA = 0	—	—
	Others	LLA: The last value is held. <sup>*1</sup>	—	—

\*1 When the alarm detection is disabled by the disable alarm detection (INH), the value is set to 0. (Page 569 Common items)

### Variation rate check (3)

- A variation rate check is performed for the duration specified by the variation rate alarm check time (CTIM). The number of executions of a variation rate check is determined by the following expression.

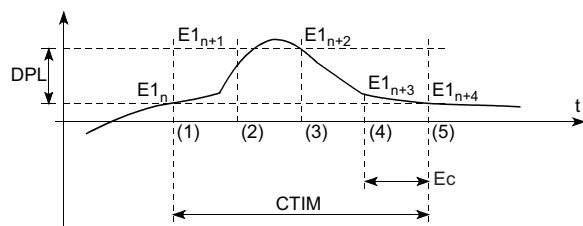
$$m = \frac{CTIM}{\Delta T}$$

Set CTIM and ΔT so that m ≥ 2.

No processing is performed when m = 0 (integral part).

**Ex.**

When m = 4, operations are performed as shown below.



- (1) 0th time: E1<sub>n</sub>-E1<sub>n-4</sub>
  - (2) 1st time: E1<sub>n</sub>-E1<sub>n-4</sub>
  - (3) 2nd time: E1<sub>n</sub>-E1<sub>n-4</sub>
  - (4) 3rd time: E1<sub>n</sub>-E1<sub>n-4</sub>
  - (5) 4th time: E1<sub>n+4</sub>-E1<sub>n</sub>
- Ec: Execution cycle

- A change in the input data and variation rate alarm value (DPL) are compared every execution cycle ( $\Delta T$ ).

Check item	Condition	ALM	BB4	BB5
Variation rate check	$E1_{n+m} - E1_n \geq DPL$	DPPA = 1*1	1*1	—
	Others	DPPA = 0	0	—
	$E1_{n+m} - E1_n \leq -DPL$	DPNA = 1*1	—	1*1
	Others	DPNA = 0	—	0

\*1 When the alarm detection is disabled by the disable alarm detection (INH), the value is set to 0. (Page 569 Common items)

#### ■Engineering value transformation (4)

The instruction performs engineering value transformation using the following expression.

$$PV = \frac{RH-RL}{100} \times E1 + RL$$

#### ■Loop stop processing (5)

The following processing is performed according to the SPA status of the alarm detection (ALM).

SPA status	Processing details
1	<p>The loop stops. When the loop stops, the following operations are performed and the S.PHPL instruction ends.</p> <ul style="list-style-type: none"> <li>• The instruction performs engineering value inverse transformation using the following expression.</li> </ul> $BW = \frac{100}{RH-RL} \times (PV-RL)$ <ul style="list-style-type: none"> <li>• The alarm bits (BB1, BB2, BB3, BB4, and BB5) are set to 0.</li> <li>• The DPNA, DPPA, LLA, HHA, PLA, and PHA of alarm detection (ALM) are set to 0.</li> </ul>
0	The loop runs and "engineering value inverse transformation (1)" is performed.

### Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1) or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Variation rate alarm value (DPL) < Variation rate alarm value (DPL)
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	Variation rate alarm check time (CTIM) < 0
	(Variation rate alarm check time (CTIM)+execution cycle ( $\Delta T$ )) > 32767

# Upper/lower limit alarm for power factor

## S.PHPL2

This instruction checks whether the input data exceeds the upper limit or underruns the lower limit for the power factor.

Ladder	ST
	ENO:=S_PHPL2(EN,s1,s2,d1,d2);

FBD/LD

### Execution condition

Instruction	Execution condition
S.PHPL2	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

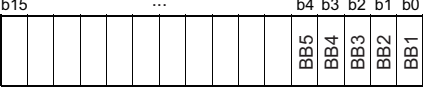
#### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□\(\H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	—	—	—	—	—	—	—	—	○	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—

#### Input data


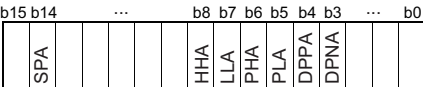
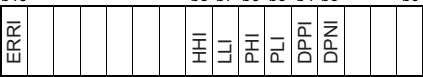

Operand: (s1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value	-999999 to 999999 [%]	Single-precision real number	—	User
+1						

## ■Block memory

Operand: (d1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999 [%]	Single-precision real number	—	System
+2	BB	BB1 Alarm BB2 Upper limit alarm BB3 Lower limit alarm BB4 Positive direction variation rate alarm BB5 Negative direction variation rate alarm	b15                    ...                    b4 b3 b2 b1 b0  0: No alarm 1: Alarm	16-bit unsigned binary	—	System

## ■Loop tag memory

The loop tag memory occupies 96 words from the specified start device.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+1	MODE	Control mode	0 to FFFFH b15                    ...                    b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 	16-bit unsigned binary	8H	User/ system
+3	ALM	Alarm detection	0 to FFFFH b15 b14                    ...                    b8 b7 b6 b5 b4 b3                    ...                    b0  SPA 0: Loop RUN 1: Loop STOP HHA, LLA, PHA, PLA, DPPA, DPNA 0: No alarm 1: Alarm	16-bit unsigned binary	4000H	User/ system
+4	INH	Disable alarm detection	0 to FFFFH b15                    ...                    b8 b7 b6 b5 b4 b3                    ...                    b0  0: Alarm detection enabled 1: Alarm detection disabled	16-bit unsigned binary	0H	User/ system
+9	N	Number of digits after the decimal point	0 to 4	16-bit unsigned binary	1	User
+10 +11	PV	Process value	TYPE = 0: (-1.0 to 1.0) TYPE = 1: (-100 to 100)  Page 675 PV output type (TYPE)	Single-precision real number	TYPE = 0: 1.0 TYPE = 1: 100	System
+26 +27	PH	Upper limit alarm value	TYPE = 0: $0.5 \leq PH \leq 1.0$ TYPE = 1: $50 \leq PH \leq 100$	Single-precision real number	TYPE = 0: 0.5 TYPE = 1: 50	User
+28 +29	PL	Lower limit alarm value	TYPE = 0: $-1.0 < PL \leq -0.5$ TYPE = 1: $-100 < PL \leq -50$	Single-precision real number	TYPE = 0: -0.5 TYPE = 1: -50	User

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+30 +31	HH	Upper upper limit alarm value	TYPE = 0: $0.5 \leq HH \leq 1.0$ TYPE = 1: $50 \leq HH \leq 100$	Single-precision real number	TYPE = 0: 0.5 TYPE = 1: 50	User
+32 +33	LL	Lower lower limit alarm value	TYPE = 0: $-1.0 < LL \leq -0.5$ TYPE = 1: $-100 < LL \leq -50$	Single-precision real number	TYPE = 0: -0.5 TYPE = 1: -50	User
+40 +41	HS	Upper/lower limit alarm hysteresis	0 to 999999 [%]	Single-precision real number	0.0	User
+42 +43	CTIM	Variation rate alarm check time	0 to 999999 [s] Set a value within the following range. $\frac{CTIM}{\Delta T} \leq 32767$	Single-precision real number	0.0	User
+44 +45	DPL	Variation rate alarm value	0 to 100 [%]	Single-precision real number	100.0	User

**Loop tag past value memory**

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

The loop tag past value memory occupies 32 words after the loop tag memory.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+124	—	Variation rate monitoring counter initialization completion flag	—	—	—	System
+125	—	Variation rate monitoring counter (rounded off to the nearest whole number)				
+126 +127	E1 <sub>n-m</sub>	—				

**Execution cycle (ΔT)**

Set the execution cycle (ΔT) in SD816 and SD817 in single-precision real number.

**PV output type (TYPE)**

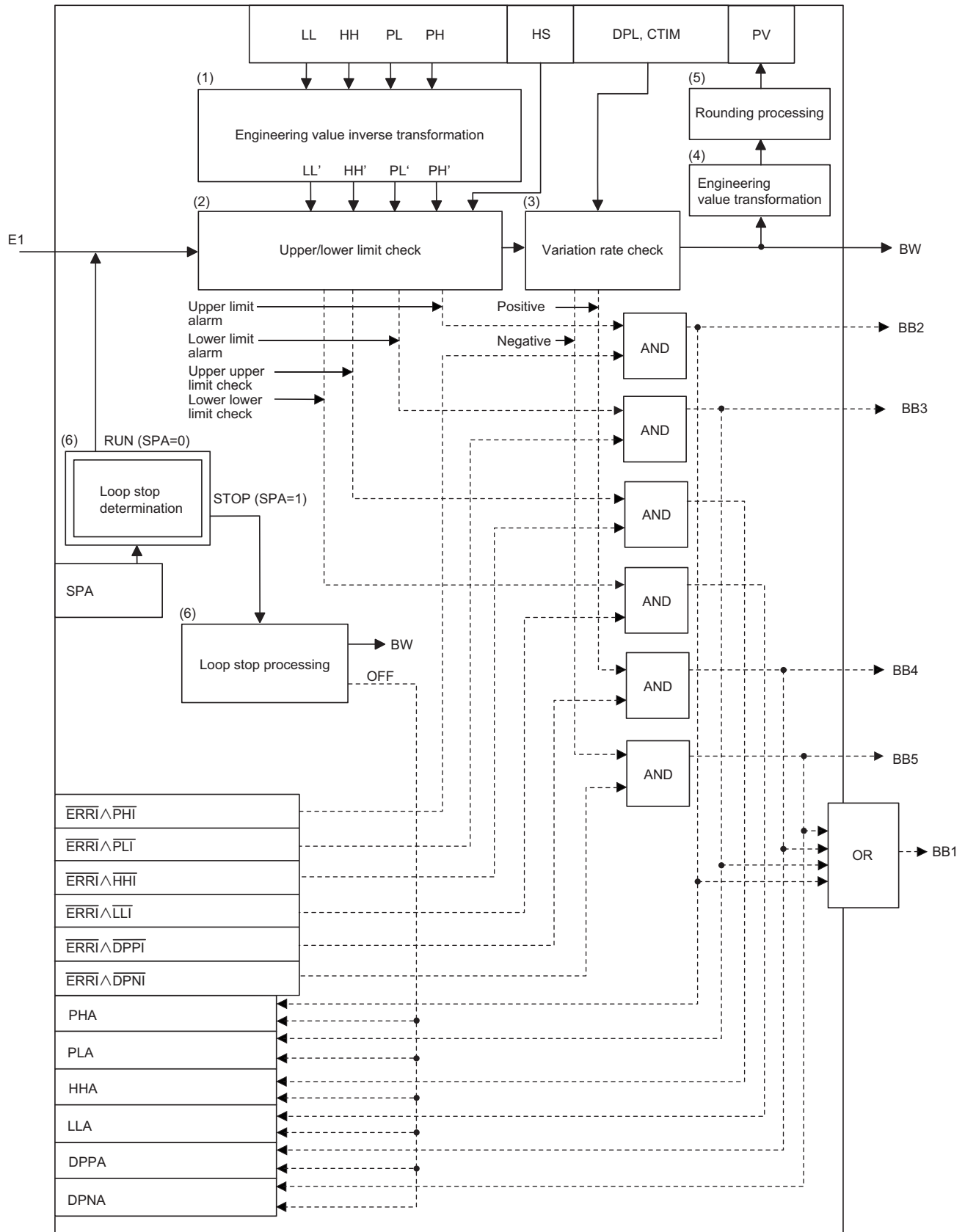
Set the PV output type (TYPE) storing 0 or 1 in SD819.

TYPE	Output type of process value (PV)
0	Decimal
1	Percentage

## Processing details

This instruction checks whether the input value (E1) exceeds the upper limit or underruns the lower limit for the power factor, and outputs an alarm based on the result.

The following is the processing block diagram of the S.PHPL2 instruction. (The numbers (1) to (6) in the diagram indicate the order of the processing.)



### ■Engineering value inverse transformation (1)

The instruction performs the following operations to match the ranges of PH, PL, HH, and LL to the input values (E1).

TYPE	Operational expression
0	$PH' = -100 \times PH + 150$
	$PL' = -100 \times PL - 50$
	$HH' = -100 \times HH + 150$
	$LL' = -100 \times LL - 50$
1	$PH' = -PH + 150$
	$PL' = -PL - 50$
	$HH' = -HH + 150$
	$LL' = -LL - 50$

### ■Upper/lower limit check (2)

The upper and lower limits of the input value (E1) are checked under the following conditions.

Check item	Condition	ALM	BB2	BB3
Upper limit check	$E1 > PH'$	$PHA = 1^{*1}$	$1^{*1}$	—
	$E1 \leq PH' - HS$	$PHA = 0$	0	—
	Others	PHA: The last value is held.*1	Hold*1	—
Lower limit check	$E1 < PL'$	$PLA = 1^{*1}$	—	$1^{*1}$
	$E1 \geq PL' + HS$	$PLA = 0$	—	0
	Others	PLA: The last value is held.*1	—	Hold*1
Upper upper limit check	$E1 > HH'$	$HHA = 1^{*1}$	—	—
	$E1 \leq HH' - HS$	$HHA = 0$	—	—
	Others	HHA: The last value is held.*1	—	—
Lower lower limit check	$E1 < LL'$	$LLA = 1^{*1}$	—	—
	$E1 \geq LL' + HS$	$LLA = 0$	—	—
	Others	LLA: The last value is held.*1	—	—

\*1 When the alarm detection is disabled by the disable alarm detection (INH), the value is set to 0. (Page 569 Common items)

**Point**

• Setting PH and HH

PH and HH are alarm values of the lagging power factor. When TYPE = 1, the recommended range is +50 to 100. An alarm is output if the PV is out of the set range of the lagging power factor. (The power factor becomes lower.) HH is the upper upper limit alarm value. Set a fewer value than the one set to PH.

• Setting PL and LL

PL and LL are alarm values of the leading power factor. When TYPE = 1, the recommended range is -50 to -100. An alarm is output if the PV is out of the set range of the leading power factor. (The power factor becomes lower.) LL is the upper upper limit alarm value. Set a greater value than the one set to PL.

### ■ Variation rate check (3)

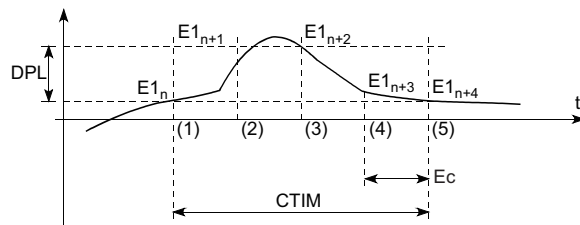
- A variation rate check is performed for the duration specified by the variation rate alarm check time (CTIM). The number of executions of a variation rate check is determined by the following expression.

$$m = \frac{CTIM}{\Delta T}$$

No processing is performed when  $m = 0$  (integral part).

**Ex.**

When  $m = 4$ , operations are performed as shown below.



- (1) 0th time:  $E1_n - E1_{n-4}$
  - (2) 1st time:  $E1_n - E1_{n-4}$
  - (3) 2nd time:  $E1_n - E1_{n-4}$
  - (4) 3rd time:  $E1_n - E1_{n-4}$
  - (5) 4th time:  $E1_{n+4} - E1_n$
- Ec: Execution cycle

- A change in the input data and variation rate alarm value (DPL) are compared every execution cycle ( $\Delta T$ ).

Check item	Condition	ALM	BB4	BB5
Variation rate check	$E1_{n+m} - E1_n \geq DPL$	DPPA = 1*1	1*1	—
	Others	DPPA = 0	0	—
	$E1_{n+m} - E1_n \leq -DPL$	DPNA = 1*1	—	1*1
	Others	DPNA = 0	—	0

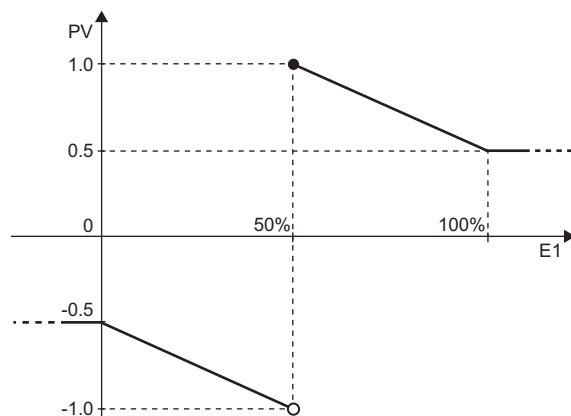
\*1 When the alarm detection is disabled by the disable alarm detection (INH), the value is set to 0. (Page 569 Common items)

### ■ Engineering value transformation (4)

The instruction performs engineering value transformation using the following expressions.

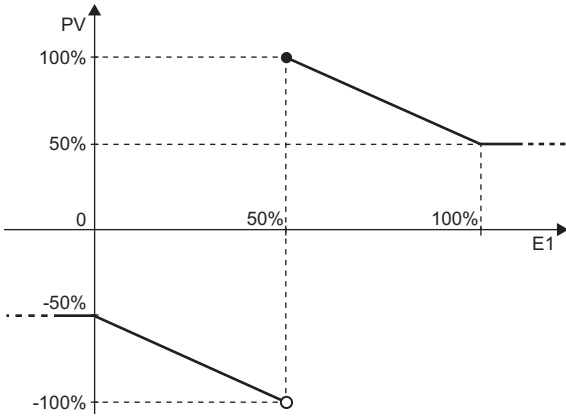
TYPE	Condition	Operational expression
0	$E1 < 0$	$PV = -0.5$
	$0 \leq E1 < 50$	$PV = -(E1 \div 100) - 0.5$
	$50 \leq E1 \leq 100$	$PV = -(E1 \div 100) + 1.5$
	$E1 > 100$	$PV = 0.5$
1	$E1 < 0$	$PV = -50$
	$0 \leq E1 < 50$	$PV = -E1 - 50$
	$50 \leq E1 \leq 100$	$PV = -E1 + 150$
	$E1 > 100$	$PV = 50$

- The relationship between the input value (E1) and process value (PV) (When TYPE = 0)





- The relationship between the input value (E1) and process value (PV) (When TYPE = 1)



### ■ Rounding the process value (PV)

Round the process value following the table below.

TYPE	N	PV	Rounded value
0	0	-0.5 or less	1.0
	1	-0.95 or less	
	2	-0.995 or less	
	3	-0.9995 or less	
	4 or more	-0.99995 or less	
1	0	-99.5 or less	100.0
	1	-99.95 or less	
	2	-99.995 or less	
	3	-99.9995 or less	
	4 or more	-99.99995 or less	

**Point**

Set an appropriate value to the number of digits after the decimal point (N) considering the PV TYPE to set in SD819.

PV has to be positive when TYPE = 1 and N = 0, based on the table above.

### ■ Loop stop processing (6)

The following processing is performed according to the SPA status of the alarm detection (ALM).

SPA status	Processing details
1	<p>The loop stops. When the loop stops, the following operations are performed and the S.PHPL2 instruction ends.</p> <ul style="list-style-type: none"> <li>• The instruction performs engineering value inverse transformation for loop stops.</li> <li>• The alarm bits (BB1, BB2, BB3, BB4, and BB5) are set to 0.</li> <li>• The DPNA, DPPA, LLA, HHA, PLA, and PHA of alarm detection (ALM) are set to 0.</li> </ul>
0	The loop runs and "engineering value inverse transformation (1)" is performed.

- Engineering value inverse transformation for loop stops

TYPE	Condition	Processing
0	$0.5 \leq PV \leq 1.0$	$BW = -100 \times (PV - 1.5)$
	$-1.0 < PV \leq -0.5$	$BW = -100 \times (PV + 0.5)$
	Others	BW: The stored value is held.
1	$50 \leq PV \leq 100$	$BW = -(PV - 150)$
	$-100 < PV \leq -50$	$BW = -(PV + 50)$
	Others	BW: The stored value is held.

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1) or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	(Variation rate alarm check time (CTIM)÷execution cycle (ΔT)) > 32767
	Variation rate alarm check time (CTIM) < 0
	Variation rate alarm value (DPL) <- Variation rate alarm value (DPL)
	The execution cycle (ΔT) setting is less than 0.

# Lead-lag compensation

## S.LLAG

This instruction performs lead-lag compensation for the input data and outputs the operation result.

Ladder	ST
	<pre>ENO:=S_LLAG(EN,s1,s2,d1,d2);</pre>

FBD/LD

### Execution condition

Instruction	Execution condition
S.LLAG	

### Setting data


#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Single-precision real number
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Local work memory start device	Refer to "Local work memory".	Single-precision real number
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

#### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□\□, U3E□\(\H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—

## Input data

Operand: (s1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Input value	-999999 to 999999 [%]	Single-precision real number	—	User
+2	e1	Actuating signal	 0: Lead-lag compensation 1: No lead-lag compensation	16-bit unsigned binary	—	User

## Block memory

Operand: (d1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999 [%]	Single-precision real number	—	System

## Operation constant

Operand: (s2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	$T_1$	Lag time	0 to 999999 [s]	Single-precision real number	1.0	User
+2 +3	$T_2$	Lead time	0 to 999999 [s]	Single-precision real number	1.0	User

## Local work memory

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

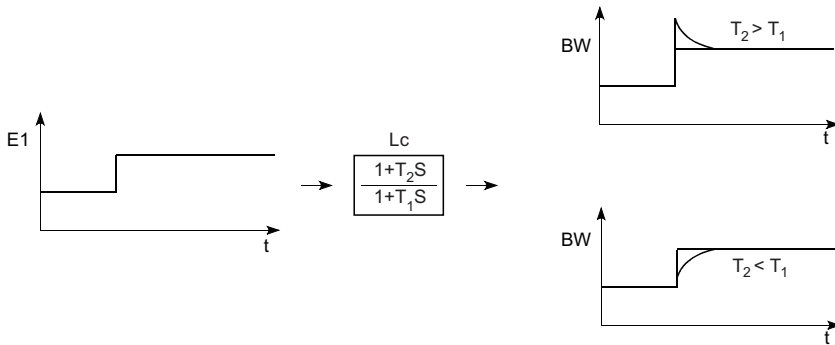
Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	$E1_{n-1}$	Last input value	—	Single-precision real number	—	System

## Execution cycle ( $\Delta T$ )

Set the execution cycle ( $\Delta T$ ) in SD816 and SD817 in single-precision real number.

## Processing details

This instruction sets the lag time ( $T_1$ ) of (s2) and the lead time ( $T_2$ ) and performs lead-lag compensation according to the actuating signal (e1).



Lc: Lead-lag compensation

The S.LLAG instruction performs the following operations.

Condition	BW (output value)
e1 = 0	$BW = \frac{1}{T_1 + \Delta T} \times \{T_2 \times (E1 - E1_{n-1}) + T_1 \times \text{Last BW value} + \Delta T \times E1\}$ <p>However, BW = 0 when <math>T + \Delta T = 0</math>.</p>
e1 = 1	BW = E1 (The input value is output as is.)

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1), (d1), (s2), or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Lag time ( $T_1$ ) < 0 or lead time ( $T_2$ ) < 0
	The execution cycle ( $\Delta T$ ) setting is less than 0.

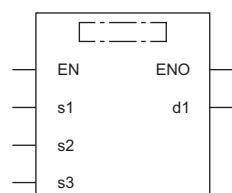
# Integral control

## S.I

This instruction performs lead-lag compensation for the input data and outputs the operation result.

Ladder	ST
	<pre>ENO:=S_I(EN,s1,s2,s3,d1);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.I	

### Setting data


### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Single-precision real number
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□\□, U3E□\□(H)G□	Z	LT, LST, LC	LZ		K	H	E	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	—	—	—	—	—	—	—	—	—	○	—

### Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Input value	-999999 to 999999	Single-precision real number	—	User
+2	e1	Operation control signal	 0: Integration operation performed 1: Integration operation not performed	16-bit unsigned binary	—	User

### Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999	Single-precision real number	—	System

### Operation constant

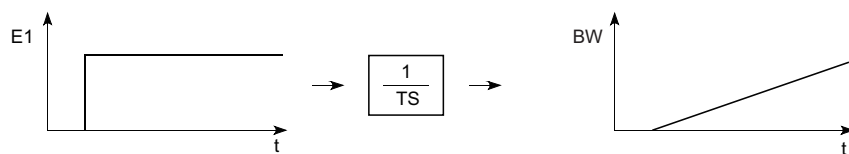
Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	T	Integral time	0 to 999999 [s]	Single-precision real number	1.0	User
+2 +3	Ys	Output initial value	-999999 to 999999	Single-precision real number	0.0	User

### Execution cycle ( $\Delta T$ )

Set the execution cycle ( $\Delta T$ ) in SD816 and SD817 in single-precision real number.

#### Processing details

This instruction performs integral operation according to the operation control signal (e1).



The S.I instruction performs the following operations.

e1	T	BW
0	≠0	$BW=Y_n = \frac{\Delta T}{T} \times E1 + Y_{n-1}$
0	0	$BW=Y_{n-1}$
1	-	$BW=Y_s$

E1: Input value of this time,  $\Delta T$ : Execution cycle,  $Y_n$ : Output value of this time,  $Y_{n-1}$ : Last output value

#### Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1), (d1), or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# Derivative control

## S.D

This instruction performs differentiation operation for the input data, and outputs the operation result.

Ladder	ST
	<pre>ENO:=S_D(EN,s1,s2,d1,d2);</pre>

FBD/LD

### Execution condition

Instruction	Execution condition
S.D	

### Setting data

#### Description, range, data type


Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Single-precision real number
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Local work memory start device	Refer to "Local work memory".	Single-precision real number
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

#### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□\□, U3E□\□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—



### Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Input value	-999999 to 999999	Single-precision real number	—	User
+2	e1	Operation control signal	 0: Differentiation operation performed 1: Differentiation operation not performed	16-bit unsigned binary	—	User

### Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999	Single-precision real number	—	System

### Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	T	Derivative time	0 to 999999 [s]	Single-precision real number	1.0	User
+2 +3	Ys	Output initial value	-999999 to 999999	Single-precision real number	0.0	User

### Local work memory

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

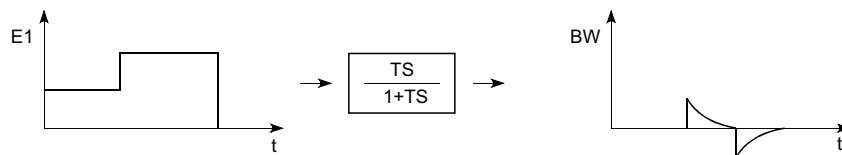
Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1 <sub>n-1</sub>	Last input value	—	Single-precision real number	—	System

### Execution cycle (ΔT)

Set the execution cycle (ΔT) in SD816 and SD817 in single-precision real number.

### Processing details

This instruction performs differentiation operation according to the operation control signal (e1).



The S.D instruction performs the following operations.

e1	BW
0	$BW = \frac{T}{T + \Delta T} \times (Y_{n-1} - E1_{n-1} + E1)$ However, BW = 0 when T + ΔT = 0.
1	BW = Ys

E1: Input value of this time, ΔT: Execution cycle, E1<sub>n-1</sub>: Last input value, Y<sub>n-1</sub>: Last output value

## Operation error

Error code (SD0)	Description
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1), (d1), (s2), or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

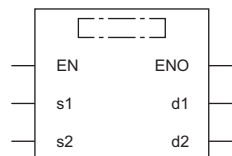
# Dead time

## S.DED

This instruction outputs the input data with the delay by the specified dead time.

Ladder	ST
	<pre>ENO:=S_DED(EN,s1,s2,d1,d2);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.DED	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Local work memory start device	Refer to "Local work memory".	Word
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

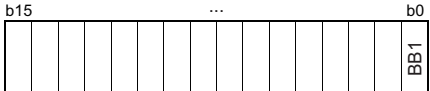
#### Applicable devices

Operand	Bit		Word				Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□\□, U3E□\□(H)G□	Z	LT, LST, LC	LZ	K, H		E	\$		
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	
(d2)	—	—	○	—	—	—	—	○	—	—	—	—	


#### Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value	-999999 to 999999	Single-precision real number	—	User
+1						
+2	e1	Operation control signal	b15 <span style="margin-left: 100px;">...</span> b0  0: Dead time 1: No dead time	16-bit unsigned binary	—	User

## ■Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999	Single-precision real number	—	System
+2	BB BB1	Data sufficiency bit	 <p>0: Data sufficient 1: Data insufficient</p>	16-bit unsigned binary	—	System

## ■Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	ST	Data sampling interval	0 to 999999 Set a value within the following range. $\frac{ST}{\Delta T} \leq 32767$	Single-precision real number	1.0	User
+2	SN	Sampling count	0 to 48	16-bit unsigned binary	0	User
+3 +4	Ys	Output initial value	-999999 to 999999	Single-precision real number	0.0	User
+5	OCHG	Initial output switching	 <p>0: Output E1 when e1 has changed from 1 to 0 SN times. 1: Output Ys up to SN.</p>	16-bit unsigned binary	0	User

## ■Local work memory

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

Operand: (d2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	—	Last value input (e1')	—	—	—	System
+1		Cycle counter*1				
+2		Number of data blocks stored in the dead time table				
+3		Dead time table 1				
+4		Dead time table 2				
+5		Dead time table 2				
+6		Dead time table 2				
⋮		⋮				
+2SN+1		Dead time table SN				
+2SN+2						

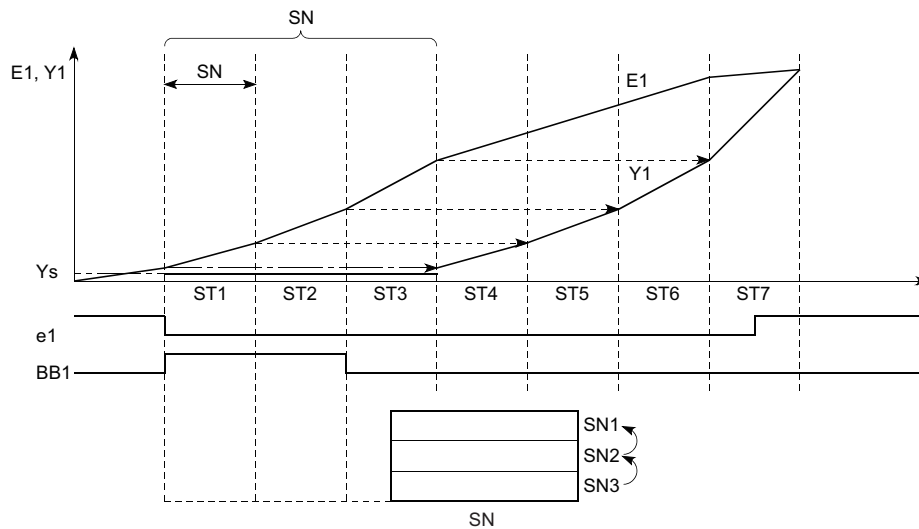
\*1 The cycle counter value is rounded off to the nearest whole number.

## ■ Execution cycle ( $\Delta T$ )

Set the execution cycle ( $\Delta T$ ) in SD816 and SD817 in single-precision real number.

### Processing details

The input value (E1) is output with a delay by the dead time according to the content of the operation control signal (e1).



SN: Dead time table

SN: Sampling count, ST: Data collection interval, E1: Input value, YS: Initial output value

The S.DED instruction performs the following operations.

e1	OCHG	Dead time	BW	
1	0/1	—	E1	
1→0	0	ST×SN	Up to SN	E1 when e1 changed from 1 to 0
			After SN	Least recent data <sup>*1</sup>
	1		Up to SN	YS
			After SN	Least recent data <sup>*1</sup>
0→0	0/1	ST×SN	Least recent data <sup>*1</sup>	

\*1 Least recent data is an input value (E1) after SN.

- When the dead time table does not have sufficient data, the data sufficiency bit (BB1) is set to 1.
- When the sampling count (SN) is 0, the data sufficiency bit (BB1) is 0 and the output value (BW) equals the input value (E1).

### Operation error

Error code (SD0)	Error content
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1) or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The sampling count (SN) is less than 0 or greater than 48.
	The data sampling interval (ST) is less than 0.
	(Data sampling interval (ST)-execution cycle ( $\Delta T$ )) is greater than 32767.

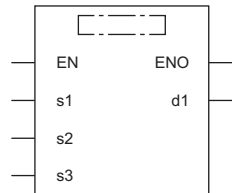
# High selector

## S.HS

This instruction outputs only the maximum value among the input data.

Ladder	ST
	ENO:=S_HS(EN,s1,s2,s3,d1);

### FBD/LD



### Execution condition

Instruction	Execution condition
S.HS	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

#### Applicable devices

Operand	Bit		Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□\□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	○	—	—	—	—	
(s2)	—	—	—	—	—	—	—	—	—	○	—	
(s3)	—	—	—	—	—	—	—	—	—	○	—	

## Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	n	Number of inputs	1 to 16	16-bit unsigned binary	—	User
+1	E1	Input value 1	-999999 to 999999	Single-precision real number	—	User
+2		Input value 2				
+3	E2	Input value 2				
+4						
⋮	⋮	⋮				
+2n-1	En	Input value n				
+2n						

## Block memory

Operand: (d1)																						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by																
+0	BW	Output value	Maximum value in E1 to En	Single-precision real number	—	System																
+1																						
+2	BB	BB1 to BB16	Output selection	16-bit unsigned binary	—	System																
			b15b14b13b12b11b10b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 <table border="1" style="margin-left: 20px;"> <tr> <td>BB16</td><td>BB15</td><td>BB14</td><td>BB13</td><td>BB12</td><td>BB11</td><td>BB10</td><td>BB9</td><td>BB8</td><td>BB7</td><td>BB6</td><td>BB5</td><td>BB4</td><td>BB3</td><td>BB2</td><td>BB1</td> </tr> </table> 0: Corresponding input value is not the maximum value. 1: Corresponding input value is the maximum value.	BB16	BB15	BB14	BB13	BB12	BB11	BB10	BB9	BB8	BB7	BB6	BB5	BB4	BB3	BB2	BB1			
BB16	BB15	BB14	BB13	BB12	BB11	BB10	BB9	BB8	BB7	BB6	BB5	BB4	BB3	BB2	BB1							

## Processing details

This instruction outputs the maximum value out of input values E1 to En.

### High selector processing

The maximum value out of input values E1 to En is stored in the output value (BW).

In addition, the BB output selection (BB1 to BB16) corresponding to the maximum value is set to 1.

The correspondence between input values 1 (E1) to 16 (E16) and BB output selections (BB1 to BB16) is shown below.

Input value	E16	E15	E14	to	E2	E1
Bit to be set to 1 at the maximum value	BB16	BB15	BB14	to	BB2	BB1

Condition	Processing	
Two or more maximum values exist.	The bits corresponding to the maximum values are all set to 1.	
Only one input	Only input value 1 (E1) is used as the input value.	<ul style="list-style-type: none"> <li>The input value 1 (E1) is stored in the output value (BW).</li> <li>BB output selection BB1 is set to 1.</li> <li>BB output selections BB2 to BB16 are set to 0.</li> </ul>
	Only one of input values 2 (E2) to 16 (E16) is used as the input value.	A value out of input values 2 (E2) to 16 (E16) and the value of input value 1 (E1) are used for data processing.

## Operation error

Error code	Error content
3402H	(s1) is a subnormal number or NaN (not a number).
3405H	The number of inputs (n) is less than 1 or greater than 16.

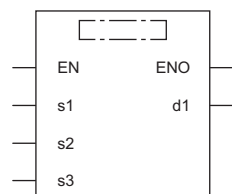
# Low selector

## S.LS

This instruction outputs only the minimum value among the input data.

Ladder	ST
	<pre>ENO:=S_LS(EN,s1,s2,s3,d1);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.LS	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

#### Applicable devices

Operand	Bit		Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	○	—	—	—	—	
(s2)	—	—	—	—	—	—	—	—	—	○	—	
(s3)	—	—	—	—	—	—	—	—	—	○	—	



### Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	n	Number of inputs	1 to 16	16-bit unsigned binary	—	User
+1 +2	E1	Input value 1	-999999 to 999999	Single-precision real number	—	User
+3 +4	E2	Input value 2				
⋮	⋮	⋮				
+2n-1 +2n	En	Input value n				

### Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	Minimum value in E1 to En	Single-precision real number	—	System
+2	BB	BB1 to BB16	Output selection	16-bit unsigned binary	—	System

b15b14b13b12b11b10b9 b8 b7 b6 b5 b4 b3 b2 b1 b0

BB16	BB15	BB14	BB13	BB12	BB11	BB10	BB9	BB8	BB7	BB6	BB5	BB4	BB3	BB2	BB1
------	------	------	------	------	------	------	-----	-----	-----	-----	-----	-----	-----	-----	-----

0: Corresponding input value is not the minimum value.  
1: Corresponding input value is the minimum value.

### Processing details

This instruction outputs the minimum value out of input values E1 to En.

#### Low selector processing

The minimum value out of input values E1 to En is stored in the output value (BW).

In addition, the BB output selection (BB1 to BB16) corresponding to the minimum value is set to 1.

The correspondence between input values 1 (E1) to 16 (E16) and BB output selections (BB1 to BB16) is shown below.

Input value	E16	E15	E14	to	E2	E1
Bit to be set to 1 at the minimum value	BB16	BB15	BB14	to	BB2	BB1

Condition	Processing	
Two or more minimum values exist.	The bits corresponding to the minimum values are all set to 1.	
Only one input	Only input value 1 (E1) is used as the input value.	<ul style="list-style-type: none"> <li>The input value 1 (E1) is stored in the output value (BW).</li> <li>BB output selection BB1 is set to 1.</li> <li>BB output selections BB2 to BB16 are set to 0.</li> </ul>
	Only one of input values 2 (E2) to 16 (E16) is used as the input value.	A value out of input values 2 (E2) to 16 (E16) and the value of input value 1 (E1) are used for data processing.

### Operation error

Error code	Error content
3402H	(s1) is a subnormal number or NaN (not a number).
3405H	The number of inputs (n) is less than 1 or greater than 16.

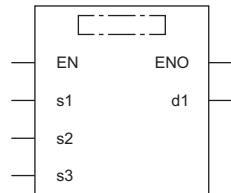
# Middle value selector

## S.MID

This instruction outputs the intermediate values between the maximum and minimum values among the input data.

ST	Ladder
	<pre>ENO:=S_MID(EN,s1,s2,s3,d1);</pre>

## FBD/LD



## Execution condition

Instruction	Execution condition
S.MID	

## Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (.).)	—	String
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (.).)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit


### Applicable devices

Operand	Bit		Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□\(\H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	○	—	—	—	—	
(s2)	—	—	—	—	—	—	—	—	—	○	—	
(s3)	—	—	—	—	—	—	—	—	—	○	—	

### Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	n	Number of inputs	1 to 16	16-bit unsigned binary	—	User
+1	E1	Input value 1	-999999 to 999999	Single-precision real number	—	User
+2		Input value 2				
+3	E2	Input value 2				
+4						
⋮	⋮	⋮				
+2n-1	En	Input value n				
+2n						

### Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	BW	Output value	Intermediate value between maximum and minimum values	Single-precision real number	—	System
+1						
+2	BB	BB1 to BB16	Output selection	16-bit unsigned binary	—	System
			b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 			
			0: Corresponding input value is not the intermediate value. 1: Corresponding input value is the intermediate value.			

### Processing details

This instruction outputs an intermediate value between the maximum and minimum values among input values E1 to En.

#### Middle value selector processing

An intermediate value among input values E1 to En is stored in the output value (BW).

In addition, the BB output selection (BB1 to BB16) corresponding to the intermediate value is set to 1.

The correspondence between input values 1 (E1) to 16 (E16) and BB output selections (BB1 to BB16) is shown below.

Input value	E16	E15	E14	to	E2	E1
Bit to be set to 1 at an intermediate value	BB16	BB15	BB14	to	BB2	BB1

- When the number of inputs is an even number, the smaller value among the intermediate values is stored.
- When two or more intermediate values exist, the bits corresponding to the intermediate values are all set to 1.

#### Remarks

Intermediate values are selected as follows:

- Input values 1 (E1) to n (En) are sorted in ascending order. (When there are same input values, they are sorted in ascending order of their input numbers.)
- The intermediate value in the sorted values is selected.

#### Ex.

When the input data are 2, 5, 1, 4, and 3, operations are performed as shown below.

In this example, 3 is the intermediate value and accordingly the output select (BB5) is set to 1.



- (1) Input data
- (2) Rearrangement
- (3) Rearranged data

## Operation error

Error code	Error content
3402H	(s1) is a subnormal number or NaN (not a number).
3405H	The number of inputs (n) is less than 1 or greater than 16.

# Average value calculation

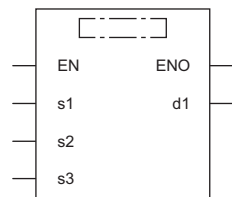
## S.AVE

Calculates and outputs the mean value of the input data.

### Execution condition

Ladder	ST
	<pre>ENO:=S_AVE(EN,s1,s2,s3,d1);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.AVE	

## Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Single-precision real number
(s2)	Empty string specification (Specify the empty string by enclosing it in double quotes (") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes (") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

### Applicable devices

Operand	Bit		Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□, U3E□\(\H)G□	Z	LT, LST, LC	LZ		K	H	E	
(s1)	—	—	○	—	—	—	○	—	—	—	—	—
(d1)	—	—	○	—	—	—	○	—	—	—	—	—
(s2)	—	—	—	—	—	—	—	—	—	○	—	—
(s3)	—	—	—	—	—	—	—	—	—	○	—	—

## Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	n	Number of inputs	1 to 16	16-bit unsigned binary	—	User
+1	E1	Input value 1	-999999 to 999999	Single-precision real number	—	User
+2						
+3	E2	Input value 2				
+4						
⋮	⋮	⋮				
+2n-1	En	Input value n				
+2n						

## Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	BW	Output value	Mean value of E1 to En	Single-precision real number	—	System
+1						

## Processing details

This instruction calculates and outputs the mean value of input values E1 to En.

### Average value calculation

The instruction calculates the mean value of input values E1 to En.

The numerical value specified by the number of inputs (n) is used as the denominator.

$$BW = \frac{E1 + E2 + E3 \dots En}{N}$$

## Operation error

Error code	Error content
3400H	An invalid operation (such as division by zero) is performed.
3402H	(s1) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The number of inputs (n) is less than 1 or greater than 16.

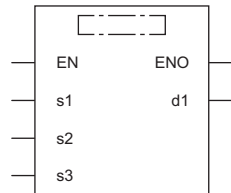
# Upper/lower limiter

## S.LIMIT

This instruction applies a limiter with hysteresis to the output value.

Ladder	ST
	<pre>ENO:=S_LIMIT(EN,s1,s2,s3,d1);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.LIMIT	

### Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit


### Applicable devices

Operand	Bit		Word		Z	Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□(H)G□		LT, LST, LC	LZ		K, H, E, \$			
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	—	—	—	—	—	—	—	—	—	○	—

### Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value	-999999 to 999999 [%]	Single-precision real number	—	User
+1						

## ■Block memory

Operand: (d1)							
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by	
+0 +1	BW	Output value	-999999 to 999999 [%]	Single-precision real number	—	System	
+2	BB	BB1	Upper limit alarm		16-bit unsigned binary	—	System
		BB2	Lower limit alarm				
			0: No alarm 1: Alarm				

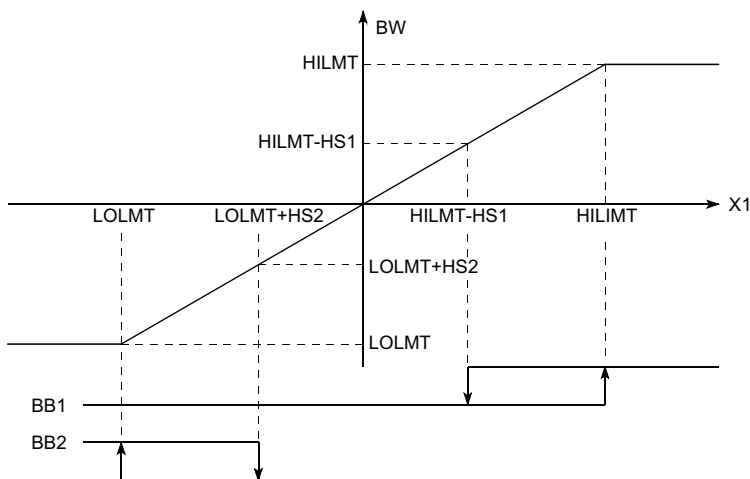
## ■Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	HILMT	Upper limit value <sup>*1</sup>	-999999 to 999999 [%]	Single-precision real number	100.0	User
+2 +3	LOLMT	Lower limit value <sup>*1</sup>	-999999 to 999999 [%]	Single-precision real number	0.0	User
+4 +5	HS1	Upper limit hysteresis	0 to 999999 [%]	Single-precision real number	0.0	User
+6 +7	HS2	Lower limit hysteresis	0 to 999999 [%]	Single-precision real number	0.0	User

\*1 Set the upper limit (HILMT) and lower limit values (LOLMT) in such a way that HILMT equals to or exceeds LOLMT.

## Processing details

This instruction applies upper and lower limiters with hysteresis to the output value.



The S.LIMIT instruction performs the following operations.

Condition	BW	BB1	BB2
$E1 \geq \text{HILMT}$	HILMT	1	0
$(\text{LOLMT} + \text{HS2}) < E1 < (\text{HILMT} - \text{HS1})$	E1	0	0
$E1 \leq \text{LOLMT}$	LOLMT	0	1
Other than the above (hysteresis part)	E1	Last value	Last value



## Operation error

Error code	Error content
3402H	The value specified by (s1) or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Upper hysteresis (HS1) is greater than 0 or the lower hysteresis (HS2) is smaller than 0.
	Lower limit value (LOLMT) is greater than upper limit value (HILMT).

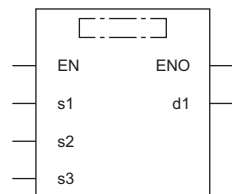
# Variation rate limiter 1

## S.VLMT1

This instruction limits the varying speed and outputs it when the variation rate of input (E1) exceeds the limit.

Ladder	ST
	<pre>ENO:=S_VLMT1(EN,s1,s2,s3,d1);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.VLMT1	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit


#### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	—	—	—	—	—	—	—	—	—	○	—

#### Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value	-999999 to 999999 [%]	Single-precision real number	—	User
+1						

## ■Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999 [%]	Single-precision real number	—	System
+2	BB	BB1 Positive direction limit alarm	 0: No alarm 1: Alarm	16-bit unsigned binary	—	System
		BB2 Negative direction limit alarm				

## ■Operation constant

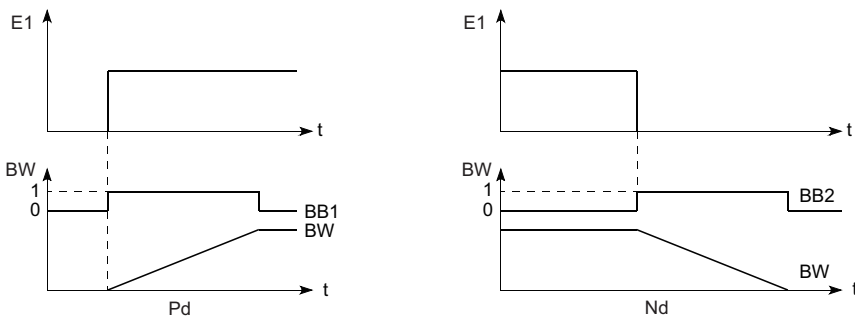
Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	V1	Positive direction limit value	0 to 999999 [%/s]	Single-precision real number	100.0	User
+2 +3	V2	Negative direction limit value	0 to 999999 [%/s]	Single-precision real number	100.0	User
+4 +5	HS1	Positive direction hysteresis	0 to 999999 [%]	Single-precision real number	0.0	User
+6 +7	HS2	Negative direction hysteresis	0 to 999999 [%]	Single-precision real number	0.0	User

## ■Execution cycle ( $\Delta T$ )

Set the execution cycle ( $\Delta T$ ) in SD816 and SD817 in single-precision real number.

### Processing details

This instruction limits the varying speed and outputs it (BW) when the variation rate of input (E1) exceeds the limit.



Pd: Positive direction  
Nd: Negative direction

The S.VLMT1 instruction performs the following operations.

Condition	Input (E1-BW)	BW	BB1	BB2
Positive direction $E1 \geq BW$	$(E1-BW) \geq (V1 \times \Delta T)$	$BW = BW + V1 \times \Delta T$	1	0
	$(E1-BW) < (V1 \times \Delta T - HS1)$	$BW = E1$	0	0
	Others	$BW = E1$	Last value	Last value
Negative direction $E1 < BW$	$(BW-E1) \geq (V2 \times \Delta T)$	$BW = BW - V2 \times \Delta T$	0	1
	$(BW-E1) < (V2 \times \Delta T - HS2)$	$BW = E1$	0	0
	Others	$BW = E1$	Last value	Last value

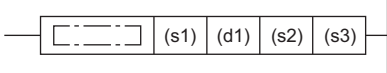
## Operation error

Error code	Error content
3402H	The value specified by (s1) or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Positive direction hysteresis (HS1) is less than 0 or the negative direction hysteresis (HS2) is less than 0.

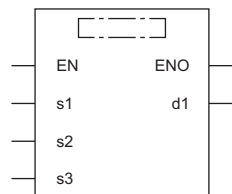
# Variation rate limiter 2

## S.VLMT2


This instruction holds the last value and outputs it when the variation rate of input (E1) exceeds the limit.

Ladder	ST
	ENO:=S_VLMT2(EN,s1,s2,s3,d1);

### FBD/LD



### ■ Execution condition

Instruction	Execution condition
S.VLMT2	

### Setting data

### ■ Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit


### ■ Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant				Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$		
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	
(s3)	—	—	—	—	—	—	—	—	—	—	○	—	

### ■ Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value	-999999 to 999999 [%]	Single-precision real number	—	User
+1						

## ■Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999 [%]	Single-precision real number	—	System
+2	BB	BB1 Positive direction limit alarm	 0: No alarm 1: Alarm	16-bit unsigned binary	—	System
		BB2 Negative direction limit alarm				

## ■Operation constant

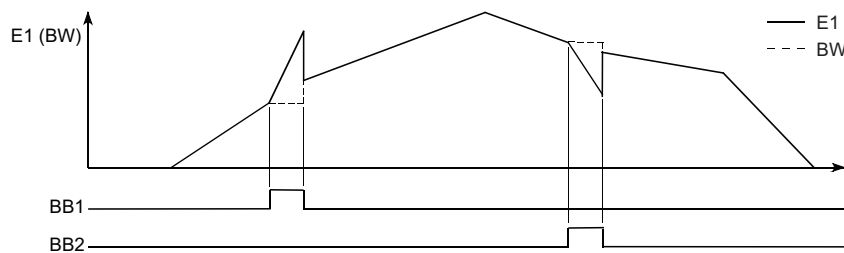
Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	V1	Positive direction limit value	0 to 999999 [%/s]	Single-precision real number	100.0	User
+2 +3	V2	Negative direction limit value	0 to 999999 [%/s]	Single-precision real number	100.0	User
+4 +5	HS1	Positive direction hysteresis	0 to 999999 [%]	Single-precision real number	0.0	User
+6 +7	HS2	Negative direction hysteresis	0 to 999999 [%]	Single-precision real number	0.0	User

## ■Execution cycle ( $\Delta T$ )

Set the execution cycle ( $\Delta T$ ) in SD816 and SD817 in single-precision real number.

### Processing details

This instruction holds the last value and outputs it when the variation rate of input (E1) exceeds the limit.



E1: Input value, BW: Output value

The S.VLMT2 instruction performs the following operations.

Condition	Input (E1-BW)	BW	BB1	BB2
Positive direction $E1 \geq BW$	$(E1-BW) \geq (V1 \times \Delta T)$	$BW=BW$	1	0
	$(E1-BW) < (V1 \times \Delta T - HS1)$	$BW=E1$	0	0
	Others	$BW=BW$	Last value	Last value
Negative direction $E1 < BW$	$(BW-E1) \geq (V2 \times \Delta T)$	$BW=BW$	0	1
	$(BW-E1) < (V2 \times \Delta T - HS2)$	$BW=E1$	0	0
	Others	$BW=BW$	Last value	Last value

## Operation error

Error code	Error content
3402H	The value specified by (s1) or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Positive direction hysteresis (HS1) is less than 0 or the negative direction hysteresis (HS2) is less than 0.

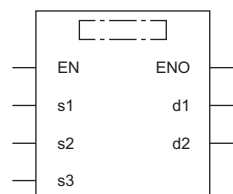
# Two-position (on/off) control

## S.ONF2

The instruction performs the following steps: SV setting, tracking, MV correction, MV output, and two-position (on/off) control.

Ladder	ST
	<pre>ENO:=S_ONF2(EN,s1,s2,s3,d1,d2);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.ONF2	

### Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
(s3)	Set value start device when E2 is used Dummy device when E2 is not used	Refer to "Set value".	Single-precision real number
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

### Applicable devices


Operand	Bit		Word		Double word			Indirect specification	Constant				Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□V□, J□□□, U3E□(H)G□	Z	LT, LST, LC	LZ		K	H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—	—
(s3)	—	—	○	—	—	—	—	○	—	—	—	—	—




## Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Input value	-999999 to 999999 [%]	Single-precision real number	—	User/ system

## Block memory



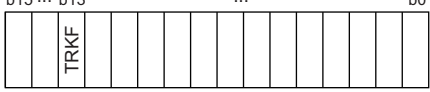
Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999 [%]	Single-precision real number	—	System
+2	BB BB1	Operation result	 <p>0:  BW &lt;50% 1:  BW ≥50%</p>	16-bit unsigned binary	—	System

## Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	PN	Operation mode	0: Reverse action 1: Direct action	16-bit unsigned binary	0	User
+1	TRK	Tracking bit	0: Tracking not performed 1: Tracking performed	16-bit unsigned binary	0	User
+2	SVPTN	Set value pattern	0 to 3  <p>(2) (1)</p> <p>(1) Use of set value Specify whether to use the set value (E2) or not. 0: Used 1: Not used (2) Set value pattern Specify whether to use the upper loop MV as the set value (E2) or not. 0: E2 is the upper loop MV. 1: E2 is not the upper loop MV.</p>	16-bit unsigned binary	3	User

## Loop tag memory

The loop tag memory occupies 96 words from the specified start device.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+1	MODE	Control mode	0 to FFFFH b15 ... b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 	16-bit unsigned binary	8H	User/system
+3	ALM	Alarm detection	0 to FFFFH b15b14 ... b0  SPA 0: Loop RUN 1: Loop STOP	16-bit unsigned binary	4000H	User/system
+4	INH	Disable alarm detection	0 to FFFFH b15 ... b13 ... b0  TRKF 0: Tracking not set 1: Tracking set	16-bit unsigned binary	0H	User/system
+12	MV	Manipulated value	-10 to 110 [%]	Single-precision real number	0.0	User/system
+13	SV	Set value	RL to RH	Single-precision real number	0.0	User
+14	DV	Deviation	-110 to 110 [%]	Single-precision real number	0.0	System
+15						
+16	HS0	Hysteresis	0 to 999999	Single-precision real number	0.0	User
+17						
+18	RH	Engineering value upper limit	-999999 to 999999	Single-precision real number	100.0	User
+19						
+22	RL	Engineering value lower limit	-999999 to 999999	Single-precision real number	0.0	User
+23						
+24	CT	Control cycle	0 to 999999 [s] Set a value within the following range. $\frac{CT}{\Delta T} \leq 32767$	Single-precision real number	1.0	User
+25						
+46						
+47						

## Loop tag past value memory

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

The loop tag past value memory occupies 32 words after the loop tag memory.

Operand: (d2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+96	—	Control cycle counter initialization completion flag	—	—	—	System
+97		Control cycle counter (The value is rounded off to the nearest whole number.)				

### ■Set value

The set value (E2) is valid only when b0 of the set value pattern (SVPTN) is set to 0 (Used). To use the upper loop MV as the set value (E2), specify the device (offset +12) where the manipulated value (MV) of the upper loop is set. If E2 is not used, specify a dummy device (SD820).

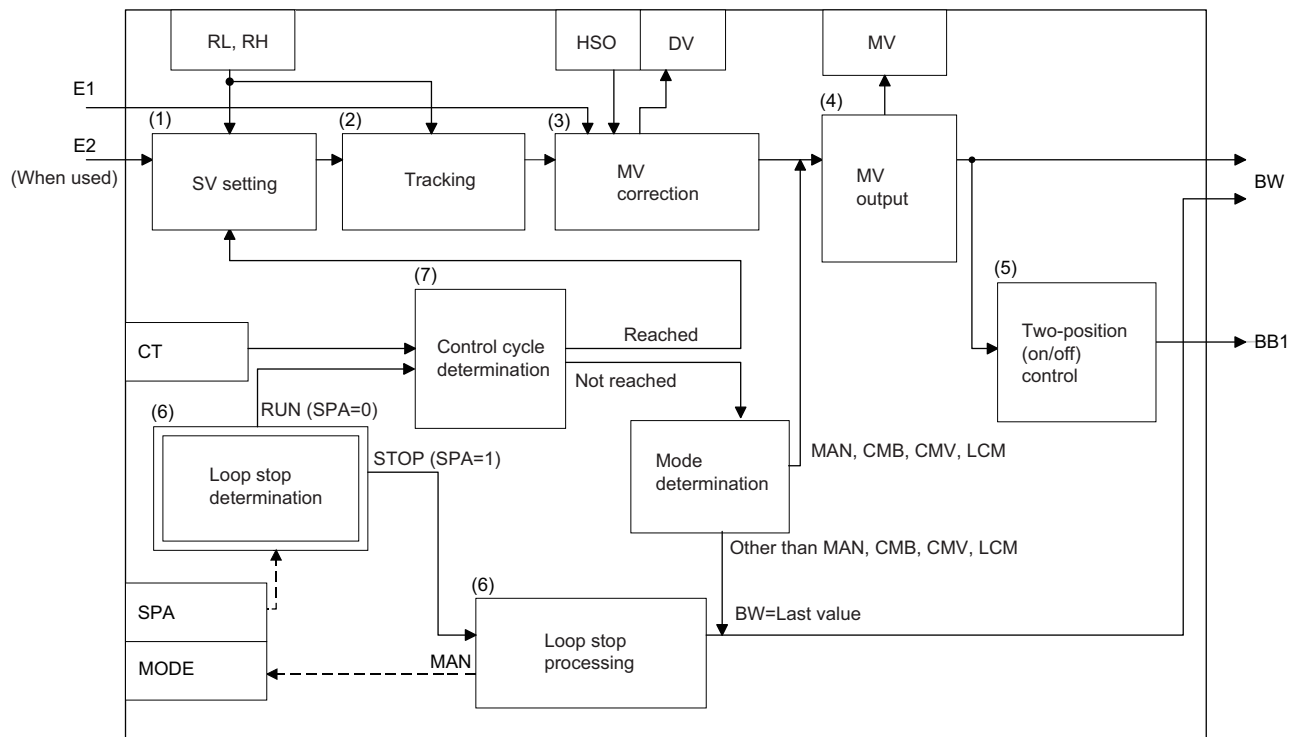
Operand: (s3)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E2	Set value	-10 to 110 [%]	Single-precision real number	0.0	User/system
+1						

### ■Execution cycle (ΔT)

Set the execution cycle (ΔT) in SD816 and SD817 in single-precision real number.

### Processing details

This instruction performs two-position (on/off) control (1-contact ON/OFF) when the specified control cycle is reached. The instruction also performs the following steps: SV setting, tracking, MV correction, and MV output processing. The following is the processing block diagram of the S.ONF2 instruction. (The numbers (1) to (7) in the diagram indicate the order of the processing.)



### ■SV setting processing (1)

The following processing is performed depending on the control mode (MODE) setting.

Control mode (MODE) setting	Processing details
CAS, CCB, CSV	<p>If the set value (E2) is specified, engineering value transformation processing (refer to the following expression) is performed, and then "tracking processing (2)" is performed.</p> $SV_n = \frac{RH-RL}{100} \times E2 + RL$ <p>If the set value (E2) is not specified, "tracking processing (2)" is performed without performing engineering value transformation processing.</p>
MAN, AUT, CMV, CMB, CAB, LCM, LCA, LCC	"Tracking processing (2)" is performed.

## ■ Tracking processing (2)

- The set value (SV) is inversely transformed from the engineering value and  $SV_n'$  is calculated (refer to the following expression).

$$SV_n' = \frac{100}{RH-RL} \times (SV_n - RL)$$

- Tracking processing is performed when all of the following conditions are satisfied.
  - The tracking bit (TRK) is set to 1.
  - The set value (E2) is used.
  - The control mode (MODE) is set to any of the following: MAN, AUT, CMV, CMB, CAB, LCM, LCA, or LCC.  
E2 =  $SV_n'$
- If the set value (E2) is the upper loop MV, the TRKF of the disable alarm detection (INH) of the upper loop is set to 1.

## ■ MV correction (3)

The instruction calculates the deviation (DV) based on the input value (E1) and the set value ( $SV_n'$ ) after tracking, and then calculates the MV correction value (MV').

- The deviation (DV) is calculated under the following conditions.

Condition	DV
Direct action (PN = 1)	$E1 - SV_n'$
Reserve action (PN = 0)	$SV_n' - E1$

- The MV correction value (MV') is calculated under the following conditions.

Condition	MV'
$DV \geq HS0$	100%
$DV \leq -HS0$	0%
$-HS0 < DV < HS0$	Last value (BW value)

## ■ MV output (4)

The manipulated value (MV (BW)) is calculated under the following conditions.

Condition	BW
CMV, MAN, CMB, LCM	$BW = MV_n$
CSV, CCB, CAB, CAS, AUT, LCC, LCA	$BW = MV'$ $MV_n = BW$

## ■ Two-position (on/off) control (5)

The BB operation result (BB1) is output under the following conditions.

Condition	BB1
$ BW  \geq 50\%$	1
$ BW  < 50\%$	0

## ■ Loop stop processing (6)

The following processing is performed according to the SPA status of the alarm detection (ALM).

SPA status	Processing details
1	The loop stops. When the loop stops, the following operations are performed and the S.ONF2 instruction ends. The output value (BW) is held. <ul style="list-style-type: none"> <li>The control mode (MODE) is set to MAN.</li> </ul> The output result (BB1) value is set to 0.
0	The loop runs and "control cycle determination processing (7)" is performed.

## ■ Control cycle determination (7)

- If the specified control cycle is not reached, the following processing is performed.

Control mode (MODE)	Processing
CSV, CCB, CAB, CAS, AUT, LCC, or LCA	The output value (BW) is held and the S.ONF2 instruction ends.
MAN, CMB, CMV, or LCM	"Three-position (on/off) control (5) is processed assuming that the output value (BW) equals the manipulated value (MV).

- If the specified control cycle is reached, "SV setting processing (1)" is performed.

## Operation error

Error code	Error content
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1) or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	Hysteresis (HS0) <- hysteresis (HS0)
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.

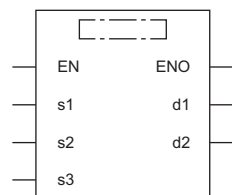
# Three-position (on/off) control

## S.ONF3

The instruction performs the following steps: SV setting, tracking, MV correction, MV output, and three-position (on/off) control.

Ladder	ST
	<pre>ENO:=S_ONF3(EN,s1,s2,s3,d1,d2);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.ONF3	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
(s3)	Set value start device when E2 is used Dummy device when E2 is not used	Refer to "Set value".	Single-precision real number
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

#### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□\□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	—	○	—	—	—	—	○	—	—	—	—




#### Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value	-999999 to 999999 [%]	Single-precision real number	—	User
+1						



## Loop tag memory

The loop tag memory occupies 96 words from the specified start device.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+1	MODE	Control mode	0 to FFFFH 	16-bit unsigned binary	8H	User/system
+3	ALM	Alarm detection	0 to FFFFH  SPA 0: Loop RUN 1: Loop STOP	16-bit unsigned binary	4000H	User/system
+4	INH	Disable alarm detection	0 to FFFFH  TRKF 0: Tracking not set 1: Tracking set	16-bit unsigned binary	0H	User/system
+12 +13	MV	Manipulated value	-10 to 110 [%]	Single-precision real number	0.0	User/system
+14 +15	SV	Set value	RL to RH	Single-precision real number	0.0	User
+16 +17	DV	Deviation	-110 to 110 [%]	Single-precision real number	0.0	System
+18 +19	HS0	Hysteresis 0	0 to 999999	Single-precision real number	0.0	User
+20 +21	HS1	Hysteresis 1	0 to 999999	Single-precision real number	0.0	User
+22 +23	RH	Engineering value upper limit	-999999 to 999999	Single-precision real number	100.0	User
+24 +25	RL	Engineering value lower limit	-999999 to 999999	Single-precision real number	0.0	User
+46 +47	CT	Control cycle	0 to 999999 [s] Set a value within the following range. $\frac{CT}{\Delta T} \leq 32767$	Single-precision real number	1.0	User

## Loop tag past value memory

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

The loop tag past value memory occupies 32 words after the loop tag memory.

Operand: (d2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+96	—	Control cycle counter initialization completion flag	—	—	—	System
+97	—	Control cycle counter (The value is rounded off to the nearest whole number.)	—	—	—	System



## ■Set value

The set value (E2) is valid only when b0 of the set value pattern (SVPTN) is set to 0 (Used). To use the upper loop MV as the set value (E2), specify the device (offset +12) where the manipulated value (MV) of the upper loop is set.

If E2 is not used, specify a dummy device (SD820).

Operand: (s3)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E2	Set value	-10 to 110 [%]	Single-precision real number	0.0	User/system
+1						

## ■Execution cycle (ΔT)

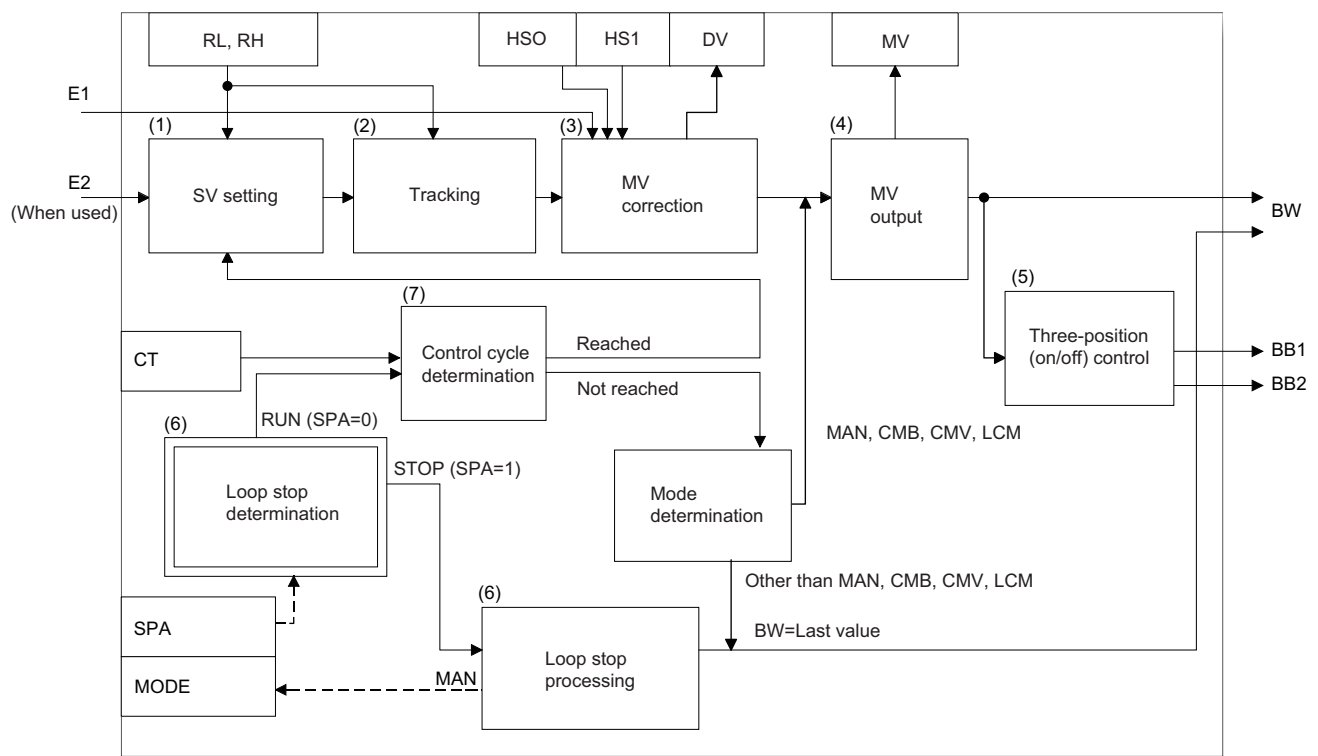
Set the execution cycle (ΔT) in SD816 and SD817 in single-precision real number.

### Processing details

This instruction performs two-position (on/off) control (2-contact ON/OFF) every control cycle.

The instruction also performs the following steps: SV setting, tracking, MV correction, and MV output processing.

The following is the processing block diagram of the S.ONF3 instruction. (The numbers (1) to (7) in the diagram indicate the order of the processing.)



## ■SV setting processing (1)

The following processing is performed depending on the control mode (MODE) setting.

Control mode (MODE) setting	Processing details
CAS, CCB, CSV	<p>If the set value (E2) is specified, engineering value transformation processing (refer to the following expression) is performed, and then "tracking processing (2)" is performed.</p> $SV_n = \frac{RH-RL}{100} \times E2 + RL$ <p>If the set value (E2) is not specified, "tracking processing (2)" is performed without performing engineering value transformation processing.</p>
MAN, AUT, CMV, CMB, CAB, LCM, LCA, LCC	"Tracking processing (2)" is performed.

## ■Tracking processing (2)

The set value (SV) is inversely transformed from the engineering value and  $SV_n'$  is calculated (refer to the following expression).

$$SV_n' = \frac{100}{RH-RL} \times (SV_n - RL)$$

Tracking processing is performed when all of the following conditions are satisfied.

- The tracking bit (TRK) is set to 1.
- The set value (E2) is used.
- The control mode (MODE) is set to any of the following: MAN, AUT, CMV, CMB, CAB, LCM, LCA, or LCC.  
E2 =  $SV_n'$
- If the set value (E2) is the upper loop MV, the TRKF of the disable alarm detection (INH) of the upper loop is set to 1.

## ■MV correction (3)

The instruction calculates the deviation (DV) based on the input value (E1) and the set value ( $SV_n'$ ) after tracking, and then calculates the MV correction value ( $MV'$ ).

- The deviation (DV) is calculated under the following conditions.

Condition	DV
Direct action (PN = 1)	$E1 - SV_n'$
Reserve action (PN = 0)	$SV_n' - E1$

- The MV correction value ( $MV'$ ) is calculated under the following conditions.

Condition	$MV'$
$DV \geq (HS1 + HS0)$	100%
$DV \leq -(HS1 + HS0)$	0%
$-(HS1 + HS0) < DV < (HS1 - HS0)$	50%
Others	Last value (BW value)

## ■MV output (4)

The manipulated value (MV (BW)) is calculated under the following conditions.

Condition	BW
CMV, MAN, CMB, LCM	$BW = MV_n$
CSV, CCB, CAB, CAS, AUT, LCC, LCA	$BW = MV'$ $MV_n = BW$

## ■Three-position (on/off) control (5)

The BB operation results (BB1, BB2) are output under the following conditions.

Condition	BB1	BB2
$BW \geq 75\%$	1	0
$25\% \leq BW < 75\%$	0	0
$BW < 25\%$	0	1

## ■Loop stop processing (6)

The following processing is performed according to the SPA status of the alarm detection (ALM).

SPA status	Processing details
1	The loop stops. When the loop stops, the following operations are performed and the S.ONF3 instruction ends. <ul style="list-style-type: none"> <li>• The output value (BW) is held.</li> <li>• The control mode (MODE) is set to MAN.</li> <li>• The output result (BB1, BB2) values are set to 0.</li> </ul>
0	The loop runs and "control cycle determination processing (7)" is performed.

## ■ Control cycle determination (7)

- If the specified control cycle is not reached, the following processing is performed.

Control mode (MODE)	Processing
CSV, CCB, CAB, CAS, AUT, LCC, or LCA	The output value (BW) is held and the S.ONF3 instruction ends.
MAN, CMB, CMV, or LCM	"Three-position (on/off) control (5) is processed assuming that the output value (BW) equals the manipulated value (MV).

- If the control cycle (CT) is reached, "SV setting processing (1)" is performed.

## Operation error

Error code	Error content
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1) or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Hysteresis 0 (HS0) is less than 0.
	The control cycle (CT) setting is less than 0.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	(Hysteresis 1 (HS1) + hysteresis 0 (HS0)) is less than 0.
	Hysteresis 1 (HS1) is less than 0.
	The value divided the control cycle (CT) by the execution cycle ( $\Delta T$ ) exceeds 32767.

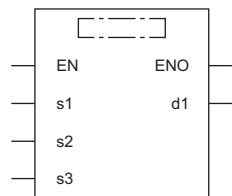
# Dead band

## S.DBND

This instruction provides a dead band and performs output processing.

Ladder	ST
	<pre>ENO:=S_DBND(EN,s1,s2,s3,d1);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.DBND	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit


#### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant				Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$		
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	
(s3)	—	—	—	—	—	—	—	—	—	—	○	—	

#### Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value	-999999 to 999999 [%]	Single-precision real number	—	User
+1						

## ■Block memory

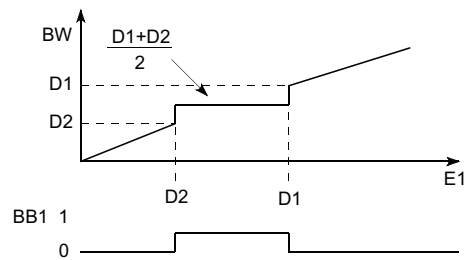
Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999 [%]	Single-precision real number	—	System
+2	BB	BB1	Dead band operation	16-bit unsigned binary	—	System
			 <p>0: Out of dead band range 1: Within dead band range</p>			

## ■Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	D1	Dead band upper limit	-999999 to 999999	Single-precision real number	100.0	User
+2 +3	D2	Dead band lower limit	-999999 to 999999	Single-precision real number	0.0	User

## Processing details

This instruction provides a dead band and performs output processing.



D1: Dead band upper limit, D2: Dead band lower limit

The S.DBND instruction performs the following operations.

Condition	BW	BB1
$D2 \leq E1 \leq D1$	$\frac{D2+D1}{2}$	1
$(E1 < D2)$ or $(E1 > D1)$	$E1$	0

## Operation error

Error code	Error content
3402H	The value specified by (s1) or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# Program setter

## S.PGS

This instruction provides control output according to the SV and MV patterns.

Ladder	ST
	<pre>ENO:=S_PGS(EN,s1,s2,d1,d2);</pre>

FBD/LD

### Execution condition

Instruction	Execution condition
S.PGS	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

#### Applicable devices

Operand	Bit		Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□□G□, J□□□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	—	—	—	—	—	—	—	—	○	—
(d1)	—	—	○	—	—	—	○	—	—	—	—	—
(s2)	—	—	—	—	—	—	—	—	—	—	○	—
(d2)	—	—	○	—	—	—	○	—	—	—	—	—



Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+54 +55	MV1	Set output 1	-10 to 110 [%]	Single-precision real number	0.0	User
⋮	⋮	⋮				
+84 +85	MV16	Set output 16				

### ■ Execution cycle ( $\Delta T$ )

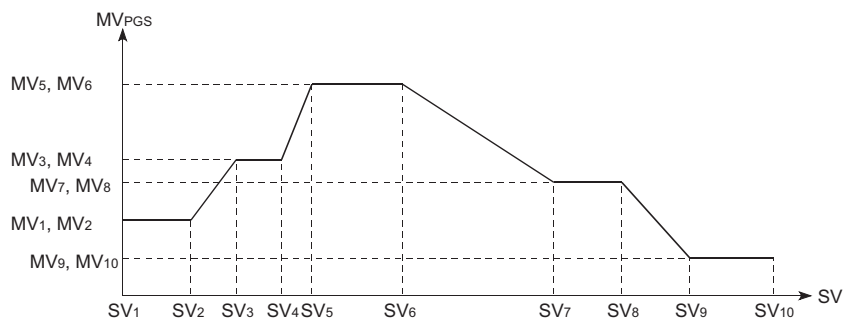
Set the execution cycle ( $\Delta T$ ) in SD816 and SD817 in single-precision real number.

### Processing details

This instruction provides control output according to the SV and MV patterns.

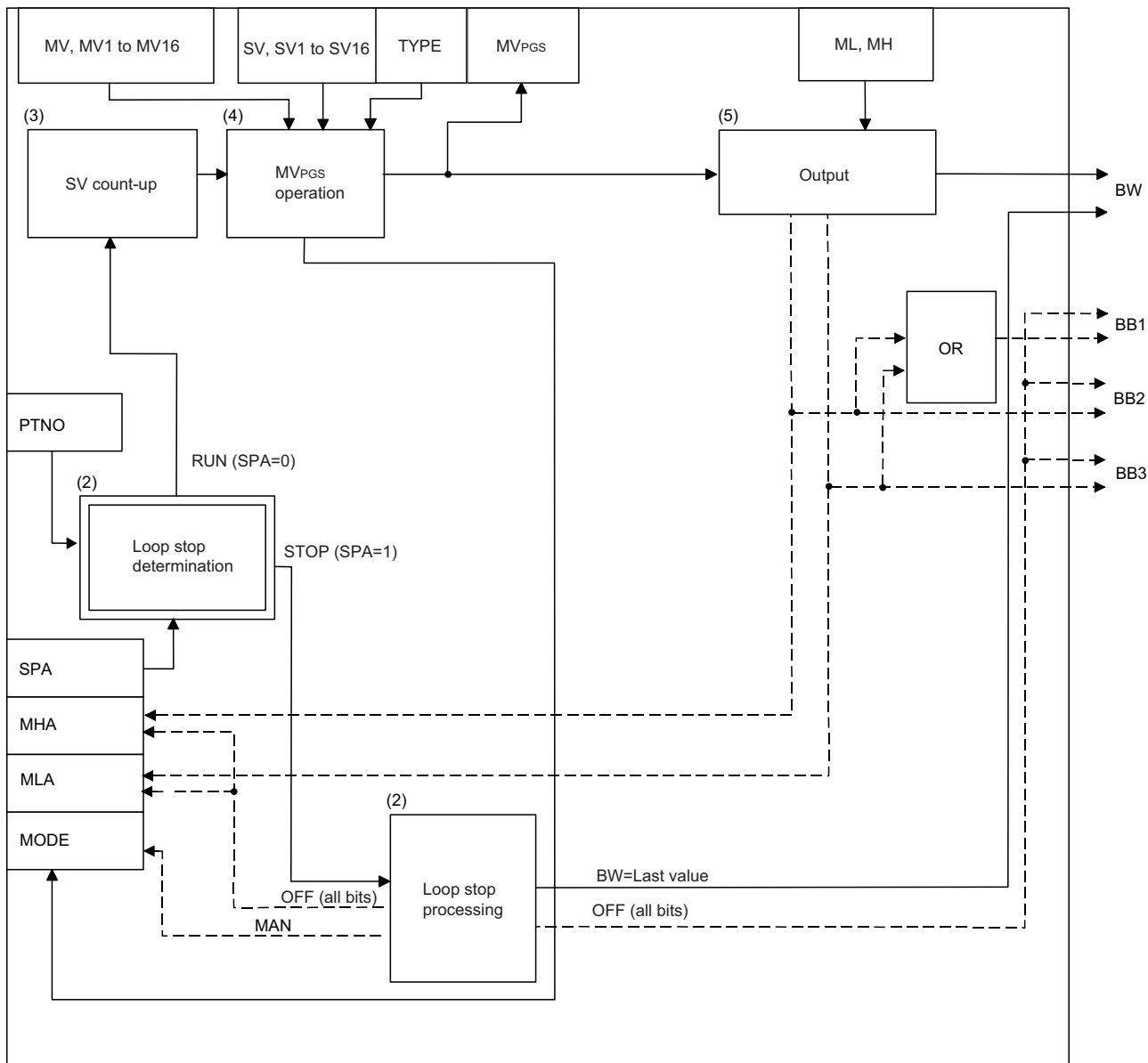
The S.PGS instruction has the following three output types.

Output type	Description
Hold type	Outputs data while holding the value of setting time 10 (SV10).
Return type	Sets the set value (SV) to 0 and outputs the last value of the manipulated value (MV).
Cyclic type	Processes based on the setting time 1 (SV1) to 10 (SV10) and then restarts processing from setting time 1 (SV1).





The following is the processing block diagram of the S.PGS instruction. (The numbers (1) to (5) in the diagram indicate the order of the processing.)



### ■Output type (1)

The output type is determined by the combination of control mode (MODE) and operation type (TYPE) as follows.

Control mode (MODE)	Operation type (TYPE)	Operation
MAN, CMB, CMV, LCM, LCA, LCC	—	Operation stop with the current SV and MV
AUT, CAB	0	Hold type operation
	1	Return type operation
CAS, CCB, CSV	—	Cyclic type operation

### ■Loop stop processing (2)

The following processing is performed according to the SPA status of the alarm detection (ALM).

SPA status	Processing details
1	<p>The loop stops.</p> <p>When the loop stops or the number of operation constant break points (PTNO) is 0, the following operations are performed and the S.PGS instruction ends.</p> <ul style="list-style-type: none"> <li>The output value (BW) is held.</li> <li>All of the output alarm bits (BB1, BB2, and BB3) are set to 0.</li> <li>The MHA and MLA of alarm detection (ALM) are set to 0.</li> <li>The control mode (MODE) is set to MAN.</li> </ul>
0	The loop runs. When the loop runs, "SV count-up processing (3)" is performed.

### ■SV count-up processing (3)

SV count-up is performed every execution cycle ( $\Delta T$ ) according to the following expression.

$$SV' = SV + \Delta T$$

### ■MVPGS operation (4)

MV<sub>PGS</sub> operation is shown below.

Output type	Hold type	Return type	Cyclic type
<b>Control mode (MODE)</b>	<b>AUT, CAB</b>		<b>CAS, CCB, CSV</b>
MV <sub>PGS</sub> operation	SV < SV1	MV1	
	SV <sub>n-1</sub> ≤ SV < SV <sub>n</sub>	$\frac{MV_n - MV_{n-1}}{SV_n - SV_{n-1}} \times (SV' - SV_{n-1}) + MV_{n-1}$	
Processing at SV' > SV <sub>n</sub>	0: Mode shift	MAN	MAN
	SV	Last value	0
	MV	Last value	Last value
	Restart method	After SV setting, change to the MAN→AUT mode.	Change to the MAN→AUT mode.

### ■Output processing (5)

The output processing conditions are shown below.

Condition	Manual			Automatic		
	MAN, CMB, CMV, LCM, LCA, LCC			AUT, CAB, CAS, CCB, CSV		
	BW	BB2, MHA	BB3, MLA	BW	BB2, MHA	BB3, MLA
MV <sub>PGS</sub> > MH	MV <sub>n</sub>	0	0	MV <sub>n</sub> = MH	1*1	0
MV <sub>PGS</sub> < ML	MV <sub>n</sub>	0	0	MV <sub>n</sub> = ML	0	1*2
Others	MV <sub>n</sub>	0	0	MV <sub>n</sub> = MV <sub>PGS</sub>	0	0

\*1 If the MHI or ERRI of the disable alarm detection (INH) is set to 1, the output upper limit alarm (BB2) and the MHA of the alarm detection (ALM) are set to 0.

\*2 If the MLI or ERRI of the disable alarm detection (INH) is set to 1, the output lower limit alarm (BB3) and the MLA of the alarm detection (ALM) are set to 0.

### Operation error

Error code	Error content
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The number of operation constant break points (PTNO) is less than 0 or greater than 16.

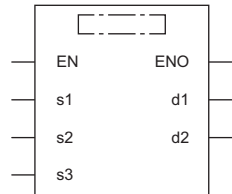
# Loop selector

## S.SEL

In automatic mode, outputs the value selected by the selection signal from the input data, and in manual mode, outputs the manipulated value (MV) in the loop tag memory.

Ladder	ST
	ENO:=S_SEL(EN,s1,s2,s3,d1,d2);

### FBD/LD



### Execution condition

Instruction	Execution condition
S.SEL	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data 1 start device	Refer to "Input data 1".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
(s3)	Input data 2 start device	Refer to "Input data 2".	Single-precision real number
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

#### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	—	○	—	—	—	—	○	—	—	—	—

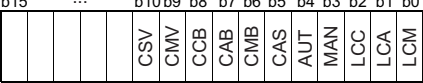

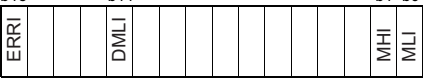
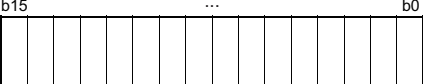
#### Input data 1

Operand: (s1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value 1	-999999 to 999999 [%]	Single-precision real number	—	User
+1						



## Loop tag memory

The loop tag memory occupies 96 words from the specified start device.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+1	MODE	Control mode	0 to FFFFH 	16-bit unsigned binary	8H	User/system
+3	ALM	Alarm detection	0 to FFFFH  <p>SPA            0: Loop RUN            1: Loop STOP            DMLA, MHA, MLA            0: No alarm            1: Alarm</p>	16-bit unsigned binary	4000H	User/system
+4	INH	Disable alarm detection	0 to FFFFH  <p>0: Alarm detection enabled            1: Alarm detection disabled</p>	16-bit unsigned binary	0H	User/system
+10	PV	Selecting a Value	RL to RH	Single-precision real number	0.0	System
+12	MV	Manipulated value	-10 to 110 [%]	Single-precision real number	0.0	User/system
+14	PV1	Process value 1	RL to RH	Single-precision real number	0.0	System
+16	PV2	Process value 2	RL to RH	Single-precision real number	0.0	System
+18	MH	Output upper limit value	-10 to 110 [%]	Single-precision real number	100.0	User
+20	ML	Output lower limit value	-10 to 110 [%]	Single-precision real number	0.0	User
+22	RH	Engineering value upper limit	-999999 to 999999	Single-precision real number	100.0	User
+24	RL	Engineering value lower limit	-999999 to 999999	Single-precision real number	0.0	User
+26	SLNO	Selected No.	 <p>(1) Selection of input value 1 (E1)            0: Not selected            1: Selected            (2) Selection of input value 2 (E2)            0: Not selected            1: Selected</p>	16-bit unsigned binary	0	System
+48	DML	Output variation rate limit value	0 to 100 [%]	Single-precision real number	100.0	User

## Input data 2

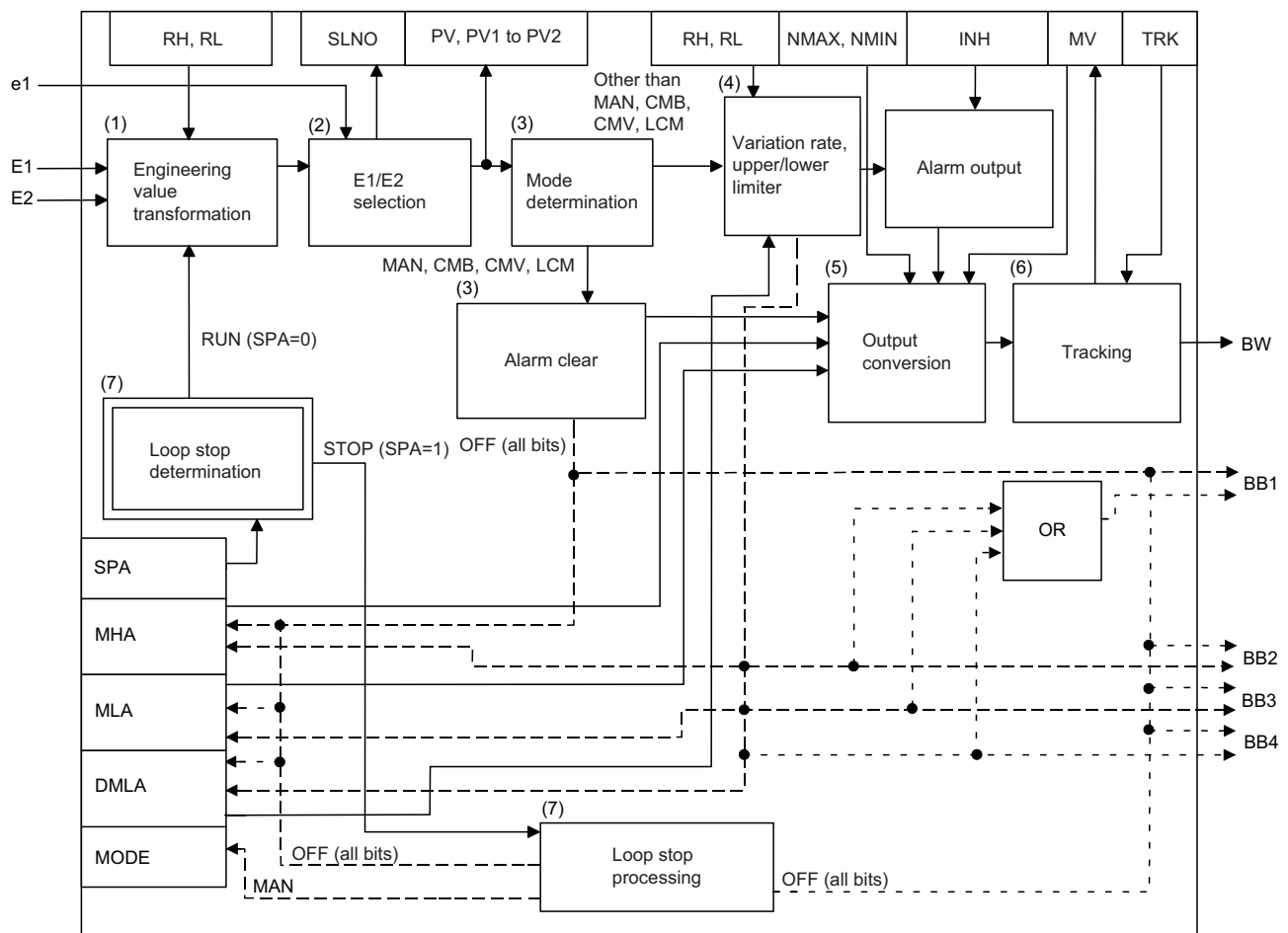
Operand: (s3)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E2	Input value 2	-999999 to 999999 [%]	Single-precision real number	0.0	User
+1						

## Processing details

This instruction outputs data in the specified mode (automatic or manual mode).

- In automatic mode, the instruction outputs the value selected by the selection signal (e1) from the input values 1 (E1) and 2 (E2).
- In manual mode, it outputs the manipulated value (MV).

The following is the processing block diagram of the S.SEL instruction. (The numbers (1) to (7) in the diagram indicate the order of the processing.)



### Engineering value transformation (1)

The instruction performs engineering value transformation using the following expression.

$$PV_n = \frac{RH-RL}{100} \times E_n + RL$$

### Input value 1 (E1), 2 (E2) selection processing (2)

Specify the input value selection (e1) of the set value pattern (SVPTN) to specify which input value is to be used, 1 (E1) or 2 (E2).

- Input value selection (e1) = 0: Input value 1 (E1) is used. Selected value (PV) = process value 1 (PV1)
- Input value selection (e1) = 1: Input value 2 (E2) is used. Selected value (PV) = process value 2 (PV2)

Selected No. (SLNO): The bit corresponding to the input value E1 or E2 is set to 1.

### ■Mode check (3)

The following processing is performed depending on the control mode (MODE).

Control mode (MODE)	Processing details
MAN, CMB, CMV, LCM	"Output conversion processing (5)" is performed. The MHA, MLA, and DMLA of the alarm detection (ALM) are set to 0. The alarm bits (BB1, BB2, BB3, and BB4) are set to 0.
AUT, CAB, CAS, CCB, CSV, LCA, LCC	The instruction performs engineering value inverse transformation using the following expression. $T = \frac{100}{RH-RL} \times (PV-RL)$ "Variation rate & upper/lower limiter processing (4)" is performed.

### ■Variation rate & upper/lower limiter (4)

The instruction checks the variation rate and upper/lower limit values for the input value 1 (E1) or 2 (E2).

The variation rate limiter conditions are shown below.

Condition	T'	BB4, DMLA
$ T-MV_n  \leq DML$	$T'=T$	0
$(T-MV_n) > DML$	$T'=MV_n+DML$	1*1
$(T-MV_n) < -DML$	$T'=MV_n-DML$	1*1

\*1 If the DMLI or ERRI of the disable alarm detection (INH) is set to 1, the output variation rate alarm (BB4) and the DMLA of the alarm detection (ALM) are set to 0.

The upper/lower limiter conditions are shown below.

Condition	MV	BB2, MHA	BB3, MLA
$T' > MH$	$MV_n = MH$	1*2	0
$T' < ML$	$MV_n = ML$	0	1*3
$ML \leq T' \leq MH$	$MV_n = T'$	0	0

\*2 If the MHI or ERRI of the disable alarm detection (INH) is set to 1, the output upper limit alarm (BB2) and the MHA of the alarm detection (ALM) are set to 0.

\*3 If the MLI or ERRI of the disable alarm detection (INH) is set to 1, the output lower limit alarm (BB3) and the MLA of the alarm detection (ALM) are set to 0.

### ■Output conversion processing (5)

The instruction performs engineering value transformation using the following expression.

$$BW = \frac{NMAX-NMIN}{100} \times MV_n + NMIN$$

### ■Tracking processing (6)

Processing is performed when the following conditions are satisfied.

Condition	Processing
When all of the following conditions are satisfied: <ul style="list-style-type: none"> <li>The control mode (MODE) is set to any of the following: MAN, CMB, CMV, or LCM.</li> <li>Tracking bit (TRK) is 1.</li> </ul> $E_n = MV_n$	Operation result is output to the input value 1 (E1) or 2 (E2).
When all of the following conditions are satisfied: <ul style="list-style-type: none"> <li>The control mode (MODE) is set to any of the following: AUT, CAS, CAB, CCB, CSV, LCA, or LCC.</li> <li>Tracking bit (TRK) is 1.</li> <li>BB alarm (BB1) = 1</li> </ul> $E_n = MV_n$	

### ■Loop stop processing (7)

The following processing is performed according to the SPA status of the alarm detection (ALM).

SPA status	Processing details
1	The loop stops. When the loop stops, the following operations are performed and the S.SEL instruction ends. <ul style="list-style-type: none"> <li>The output value (BW) is held.</li> <li>The alarm bits (BB1, BB2, BB3, and BB4) are set to 0.</li> <li>The MHA, MLA, and DMLA of the alarm detection (ALM) are set to 0.</li> <li>The control mode (MODE) is set to MAN.</li> </ul>
0	Processing from "Engineering value transformation (1)" through to "Tracking processing (6)" is performed.

## Operation error

Error code	Error content
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1), (s2), (d2), or (s3) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.



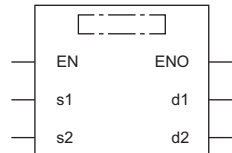
# Bumpless transfer

## S.BUMP

This instruction gradually brings the output value (BW) closer to the output set value (E1) from the output control value (E2) when the mode switching signal (e1) changes from manual to automatic.

Ladder	ST
	<pre>ENO:=S_BUMP(EN,s1,s2,d1,d2);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.BUMP	

### Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Single-precision real number
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Local work memory start device	Refer to "Local work memory".	Word
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

### Applicable devices

Operand	Bit		Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□\□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	○	—	—	—	—	
(d2)	—	—	○	—	—	—	○	—	—	—	—	

### Input data

Operand (s1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Output set value	-999999 to 999999 [%]	Single-precision real number	—	User
+1	E2	Output control value	-999999 to 999999 [%]	Single-precision real number	—	User
+4	e1	Mode switching signal	0: Manual mode 1: Automatic mode	16-bit unsigned binary	—	User

## ■Block memory

Operand (d1)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999 [%]	Single-precision real number	—	System

## ■Operation constant

Operand: (s2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	T	Lag time	0 to 999999 [s]	Single-precision real number	1.0	User
+2 +3	a	Lag band	0 to 999999 [%]	Single-precision real number	1.0	User

## ■Local work memory

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	Xq	Initial deviation value	—	Single-precision real number	1.0	System
+2 +3	Xp	Deviation				

## ■Execution cycle ( $\Delta T$ )

Set the execution cycle ( $\Delta T$ ) in SD816 and SD817 in single-precision real number.

### Processing details

This instruction gradually brings the output value (BW) closer to the output set value (E1) from the output control value (E2) when the control mode changes from manual to automatic.

When the output value (BW) enters the range specified by the lag band (a) on the basis of the output set value (E1), the instruction brings it closer to the output set value (E1) with a primary lag.

In manual mode (mode switching signal (e1) = 0), the instruction calculates the output value (BW), initial deviation value (Xq), and deviation (Xp) from the following expressions.

- Output value (BW) = output control value (E2)
- Initial deviation value (Xq) = output control value (E2) - output set value (E1)
- Deviation (Xp) = output control value (E2) - output set value (E1)

In automatic mode (mode switching signal (e1) = 1), the instruction calculates the output value (BW) from the following expressions.

Condition	$ Xp  > a$	$ Xp  \leq a$
Xp	$Xp = Xp' - \frac{\Delta T}{T} Xq$	$Xp = \frac{T}{T + \Delta T} Xp'$
BW	$BW = E1 + Xp$  $ Xp  \leq \frac{\Delta T}{T}  Xq $  On the condition above, • BW = E1 • $Xp = Xp'$	$BW = E1 + Xp$ provided that $ Xp  \leq 10^{-4}$ • BW = E1 • $Xp = Xp'$

However, in automatic mode in which lag time (T)  $\leq$  execution cycle ( $\Delta T$ ), output value (BW) = output set value (E1), deviation (Xp) = Xp'.

## Operation error

Error code	Error content
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1), (s2), or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

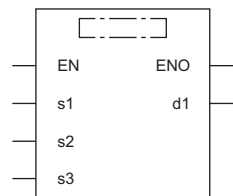
# Analog memory

## S.AMR

This instruction increases or decreases the output value (BW) at a fixed rate.

Ladder	ST
	<pre>ENO:=S_AMR(EN,s1,s2,s3,d1);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.AMR	

### Setting data

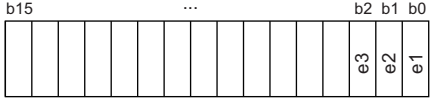
#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Single-precision real number
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

#### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	—	—	—	—	—	—	—	—	—	○	—

## Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Output addition value	-999999 to 999999	Single-precision real number	—	User
+2 +3	E2	Output subtraction value	-999999 to 999999	Single-precision real number	—	User
+4 +5	E3	Output set value	-999999 to 999999	Single-precision real number	—	User
+6	e1 e2 e3	Operation output signal Output addition signal Output subtraction signal	 <p>e1 0: Manual mode 1: Automatic mode e2 0: Do not add 1: Add e3 0: Do not subtract 1: Subtract</p>	16-bit unsigned binary	—	User

## Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999	Single-precision real number	—	System

## Operation constant

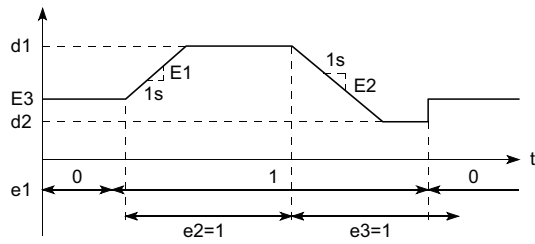
Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	d1	Output upper limit value	0 to 999999	Single-precision real number	1.0	User
+2 +3	d2	Output lower limit value	-999999 to 999999	Single-precision real number	1.0	User

## Execution cycle ( $\Delta T$ )

Set the execution cycle ( $\Delta T$ ) in SD816 and SD817 in single-precision real number.

## Processing details

This instruction increases or decreases the output value (BW) at a fixed rate.



The instruction performs the following processing as specified by operation output signal (e1), output addition signal (e2), and output subtraction signal (e3).

- In manual mode (operation output signal (e1) = 0), the output value (BW) equals the output set value (E3).
- In automatic mode (operation output signal (e1) = 1), the instruction performs the following operations as specified by the output addition signal (e2) and output subtraction signal (e3).

e2	e3	BW
1	0	$BW = BW +  E1  \times \Delta T$ provided that when $d1 \leq BW$ , BW equals d1.
0	1	$BW = BW -  E2  \times \Delta T$ provided that when $BW \leq d2$ , BW equals d2.
1	1	$BW = BW$
0	0	$BW = BW$

## Operation error

Error code	Error content
3402H	The value specified by (s1) or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

# 21.5 Correction Operation Instructions

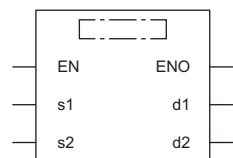
## Function generator

### S.FG

This instruction outputs the input data values following the specified function generator pattern.

Ladder	ST
	<pre>ENO:=S_FG(EN,s1,s2,d1,d2);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.FG	

### Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Single-precision real number
(s2)	Operation constant start device	Refer to "Operation constant".	16-bit unsigned binary
(d2)	Local work memory start device	Refer to "Local work memory".	Word
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

### Applicable devices

Operand	Bit		Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□\□, U3E□\□(H)G□	Z	LT, LST, LC	LZ		K, H, E, \$			
(s1)	—	—	○	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	○	○	—	—	—	
(d2)	—	—	○	—	—	—	○	—	—	—	—	

### Input data

Operand: (s1)							
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by	
+0	E1	Input value	-999999 to 999999	Single-precision real number	—	User	
+1							

## ■Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999	Single-precision real number	—	System

## ■Operation constant

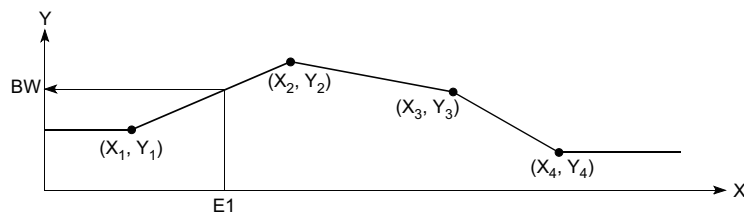
Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	SN	Number of break points	0 to 48	16-bit unsigned binary	0	User

## ■Local work memory

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	X1	Break point coordinates	-999999 to 999999	Single-precision real number	—	User
+2 +3	Y1	Break point coordinates				
+4 +5	X2	Break point coordinates				
+6 +7	Y2	Break point coordinates				
⋮	⋮	⋮				
+4SN-4 +4SN-3	Xn	Break point coordinates				
+4SN-2 +4SN-1	Yn	Break point coordinates				

## Processing details

This instruction outputs values according to the function generator pattern consisting of n break points as specified by (s2) with regard to the input value (E1).



The S.FG instruction performs the following operations.

Condition	Output value (BW)
$E1 \leq X_1$	$BW = Y_1$
$X_{i-1} < E1 \leq X_i (i=2 \text{ to } n)$	$BW = \frac{Y_i - Y_{i-1}}{X_i - X_{i-1}} \times (E1 - X_{i-1}) + Y_{i-1}$
$X_n < E1$	$BW = Y_n$

- If the value specified in (n) is 0, no processing is performed.
- If  $X_{i-1} > X_i$ , processing is stopped when  $n = i - 1$ . (The subsequent data is ignored.)
- When there are two or more  $X_i$  for the same  $Y_i$ , the Y with smaller i is selected.



## Operation error

Error code	Error content
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1) or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The number of break points (SN) is less than 0 or greater than 48.

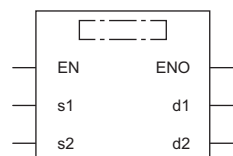
# Inverse function generator

## S.IFG

This instruction outputs the input data values following the specified inverse function generator pattern.

Ladder	ST
	<pre>ENO:=S_IFG(EN,s1,s2,d1,d2);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.IFG	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Single-precision real number
(s2)	Operation constant start device	Refer to "Operation constant".	16-bit unsigned binary
(d2)	Local work memory start device	Refer to "Local work memory".	Word
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

#### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□\□, U3E□\□(H)G□	Z	LT, LST, LC	LZ		K	H	E	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	○	—	—	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—

#### Input data

Operand: (s1)							
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by	
+0	E1	Input value	-999999 to 999999	Single-precision real number	—	User	
+1							

### ■Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999	Single-precision real number	—	System

### ■Operation constant

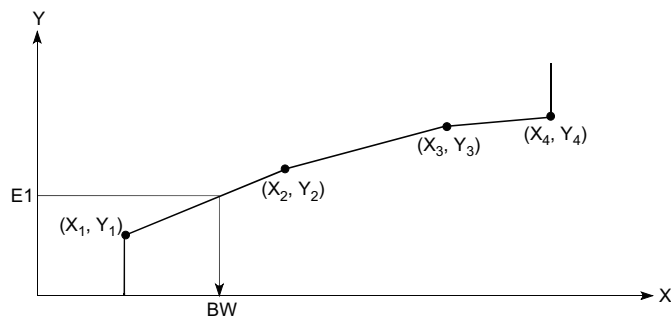
Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	SN	Number of break points	0 to 48	16-bit unsigned binary	0	User

### ■Local work memory

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	X1	Break point coordinates	-999999 to 999999	Single-precision real number	—	User
+2 +3	Y1	Break point coordinates				
+4 +5	X2	Break point coordinates				
+6 +7	Y2	Break point coordinates				
⋮	⋮	⋮				
+4SN-4 +4SN-3	Xn	Break point coordinates				
+4SN-2 +4SN-1	Yn	Break point coordinates				

### Processing details

This instruction outputs values according to the inverse function generator pattern consisting of n break points as specified by (s2) with regard to the input value (E1).



The S.IFG instruction performs the following operations.

Condition	Output value (BW)
$E1 \leq Y1$	$BW = X1$
$Y_{i-1} < E1 \leq Y_i (i=2 \text{ to } n)$	$BW = \frac{X_i - X_{i-1}}{Y_i - Y_{i-1}} \times (E1 - Y_{i-1}) + X_{i-1}$
$Y_n < E1$	$BW = X_n$

- If the value specified in (n) is 0, no processing is performed.
- If  $X_{i-1} > X_i$ , processing is stopped when  $n = i - 1$ . (The subsequent data is ignored.)
- When there are two or more  $X_i$  for the same  $Y_i$ , the X with smaller i is selected.

## Operation error

Error code	Error content
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1) or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The number of break points (SN) is less than 0 or greater than 48.

# Standard filter

## S.FLT

This function outputs the mean value of the n pieces of data sampled at the specified data collection intervals (ST).

Ladder	ST
	<pre>ENO:=S_FLT(EN,s1,s2,d1,d2);</pre>

FBD/LD

### Execution condition

Instruction	Execution condition
S.FLT	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(d2)	Local work memory start device	Refer to "Local work memory".	Word
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

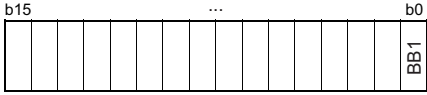
#### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant				Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□\□, U3E□\□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$		
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—	—

#### Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value	-999999 to 999999	Single-precision real number	—	User
+1						

## ■Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999	Single-precision real number	—	System
+2	BB BB1	Data sufficiency bit	 <p>0: Data sufficient 1: Data insufficient</p>	Unsigned 16-bit binary data	—	System

## ■Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	ST	Data sampling interval	0 to 999999 [s] Set a value within the following range. $\frac{ST}{\Delta T} \leq 32767$	Single-precision real number	1.0	User
+2	SN	Sampling count	0 to 48	16-bit unsigned binary	0	User

## ■Local work memory

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

Operand: (d2)						
Position	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	ST'	Last data sampling interval	—	Single-precision real number	—	System
+2	SN'	Last sampling count		16-bit unsigned binary		System
+3	i	Cycle counter*1		16-bit unsigned binary		System
+4	n1	Number of data blocks stored		16-bit unsigned binary		System
+5	n2	Storage address		16-bit unsigned binary		System
+6 +7	—	—	—	—	—	—
+8 +9	1	Dead time table 1	—	Single-precision real number	—	System
+10 +11	2	Dead time table 2				
⋮	⋮	⋮				
+2SN+6 +2SN+7	SN	Dead time table SN				

\*1 The cycle counter value is rounded off to the nearest whole number.

## ■Execution cycle (ΔT)

Set the execution cycle (ΔT) in SD816 and SD817 in single-precision real number.

## Processing details

This instruction samples input values (E1) at data sampling intervals (ST) by the number of samples (SN), stores them in the dead time table, averages the number of collected data blocks (SN), and outputs the result.

- The data refreshing cycle is "data sampling interval (ST)÷execution cycle ( $\Delta T$ )". (The resultant value is rounded off to the nearest whole number.)
- When the dead time table is filled with the number of sampled data blocks (SN), the data sufficiency bit (BB1) is set to 0. When the dead time table does not have sufficient data, BB1 is set to 1.

### Point

- Until the dead time table is filled with data, the past data is averaged and output.
- Processing is performed by  $ST=n\times\Delta T$  (n is an integer).

## Operation error

Error code	Error content
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1), (s2), or (d2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The sampling count (SN) is less than 0 or greater than 48.
	The execution cycle ( $\Delta T$ ) setting is less than 0.
	The data sampling interval (ST) is less than 0.
	(Data sampling interval (ST)÷execution cycle ( $\Delta T$ )) is greater than 32767.

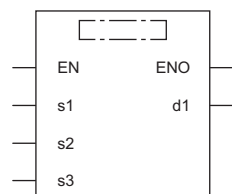
# Integration

## S.SUM

This instruction integrates and outputs the input data.

Ladder	ST
	<pre>ENO:=S_SUM(EN,s1,s2,s3,d1);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.SUM	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Single-precision real number
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

#### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	—	—	—	—	—	—	—	—	—	○	—



## Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Input value	-999999 to 999999	Single-precision real number	—	User
+2	e	e1	Integration start signal	16-bit unsigned binary	—	User

0: Integration operation not performed  
1: Integration operation performed

## Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999	Single-precision real number	—	System

## Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	ILC	Input low-cut value	-999999 to 999999	Single-precision real number	0.0	User
+2 +3	A	Initial value	-999999 to 999999	Single-precision real number	0.0	User
+4	RANGE	Input range	1: /second 2: /minute 3: /hour	16-bit unsigned binary	1	User

## Execution cycle ( $\Delta T$ )

Set the execution cycle ( $\Delta T$ ) in SD816 and SD817 in single-precision real number.

### Processing details

When the integration start signal (e1) changes from 0 to 1, this instruction integrates the input values (E1) and outputs the result.

The S.SUM instruction performs the following operations.

e1	E1	Output (BW)
0	—	Outputs the initial value (A) of the operation constant.
1	$E1 \leq ILC$	The last value is output as is.
	$E1 > ILC$	$BW = E1 \times \frac{\Delta T}{T} + \text{Last value}$

The value of T used for operation varies depending on the setting of the input range (RANGE).

- Input range (RANGE) = 1: T = 1
- Input range (RANGE) = 2: T = 60
- Input range (RANGE) = 3: T = 3600

### Operation error

Error code	Error content
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1), (d1), or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Input range (RANGE) is less than 1 or greater than 3.

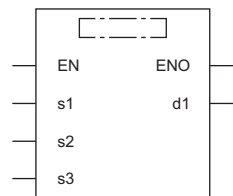
# Temperature/pressure correction

## S.TPC

This instruction outputs input data after temperature/pressure correction.

Ladder	ST
	<pre>ENO:=S_TPC(EN,s1,s2,s3,d1);</pre>

## FBD/LD



### Execution condition

Instruction	Execution condition
S.TPC	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Single-precision real number
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

#### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	—	—	—	—	—	—	—	—	—	○	—

**Input data**

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Differential pressure	-999999 to 999999	Single-precision real number	—	User
+2 +3	E2	Measured temperature	-999999 to 999999	Single-precision real number	—	User
+4 +5	E3	Measured pressure	-999999 to 999999	Single-precision real number	—	User
+6	e	e1 E2 use flag	<p>0: Not used 1: Used</p>	16-bit unsigned binary	—	User
		e2 E3 use flag				

**Block memory**

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999	Single-precision real number	—	System

**Operation constant**

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	TEMP	Designed temperature T' (engineering value)	-999999 to 999999 [°C]	Single-precision real number	0.0	User
+2 +3	B1	Bias (temperature)	-999999 to 999999 [°C]	Single-precision real number	273.15	User
+4 +5	PRES	Designed pressure P' (engineering value)	-999999 to 999999	Single-precision real number	0.0	User
+6 +7	B2	Bias (pressure)	-999999 to 999999	Single-precision real number	10332.0	User

**Processing details**

This instruction outputs the input value (E1) after performing temperature or pressure correction over it.

The S.TPC instruction calculates the temperature/pressure correction value using the following expression.

$$BW = E1 \times A1 \times A2$$

The values of A1 and A2 are calculated by the following expressions.

Input		A1	A2
e1	e2		
Used	Used	$\frac{T'+B1}{E2+B1}$	$\frac{E3+B2}{P'+B2}$
Not used	Used	1.0	$\frac{E3+B2}{P'+B2}$
Used	Not used	$\frac{T'+B1}{E2+B1}$	1.0
Not used	Not used	1.0	1.0

## Operation error

Error code	Error content
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1) or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

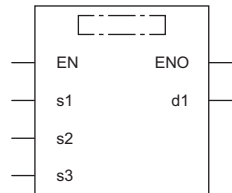
# Engineering value transformation

## S.ENG

This instruction performs engineering value inverse transformation of the input data.

Ladder	ST
	<pre>ENO:=S_ENG(EN,s1,s2,s3,d1);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.ENG	

### Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Single-precision real number
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

### Applicable devices

Operand	Bit		Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□, U3E□\□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—	—
(d1)	—	—	○	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—	—
(s3)	—	—	—	—	—	—	—	—	—	○	—	—

### Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Input value	-999999 to 999999 [%]	Single-precision real number	—	User

## ■Block memory

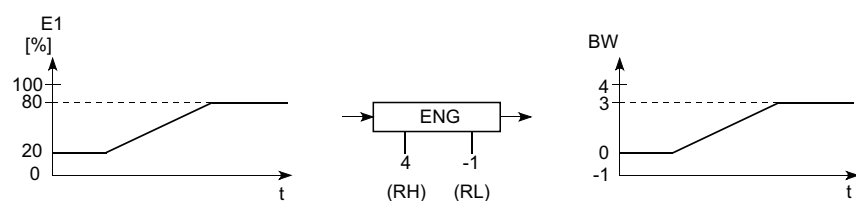
Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999	Single-precision real number	—	System

## ■Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	RH	Engineering value upper limit	-999999 to 999999	Single-precision real number	100.0	User
+2 +3	RL	Engineering value lower limit	-999999 to 999999	Single-precision real number	0.0	User

## Processing details

This instruction performs engineering value transformation of the input data (E1) and outputs it.



The S.ENG instruction performs the following operations.

$$BW = \frac{RH-RL}{100} \times E1 + RL \quad (E1=0 \text{ to } 100\%)$$

## Operation error

Error code	Error content
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1) or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

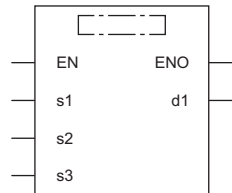
# Engineering value inverse transformation

## S.IENG

The instruction performs engineering value inverse transformation processing to the input data.

Ladder	ST
	<pre>ENO:=S_IENG(EN,s1,s2,s3,d1);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.IENG	

### Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Single-precision real number
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

### Applicable devices

Operand	Bit		Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□\(\H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—	—
(d1)	—	—	○	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—	—
(s3)	—	—	—	—	—	—	—	—	—	○	—	—

### Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Input value	-999999 to 999999	Single-precision real number	—	User

## ■Block memory

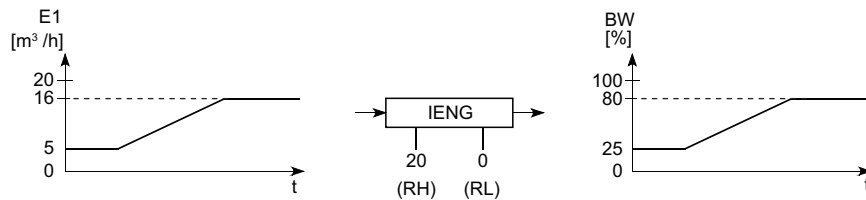
Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999 [%]	Single-precision real number	—	System

## ■Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	RH	Engineering value upper limit	-999999 to 999999	Single-precision real number	100.0	User
+2 +3	RL	Engineering value lower limit	-999999 to 999999	Single-precision real number	0.0	User

## Processing details

This instruction converts the input value (E1) to the % value and outputs it.



The S.IENG instruction performs the following operations.

$$BW = \frac{100}{RH-RL} \times (E1-RL) [\%]$$

Set the engineering value upper limit (RH) and lower limit (RL) so that RH is greater than RL.

Even when RH equals or less than RL, processing is executed accordingly but does not result in engineering value inverse transformation.

When RH equals RL, the output value (BW) becomes 0.

## Operation error

Error code	Error content
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1) or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.



# 21.6 Arithmetic Operation Instructions

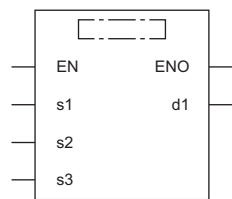
## Addition

### S.ADD

This instruction adds input data with a coefficient.

Ladder	ST
	ENO:=S_ADD(EN,s1,s2,s3,d1);

### FBD/LD



### Execution condition

Instruction	Execution condition
S.ADD	

### Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Single-precision real number
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□\□, U3E□\□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	—	—	—	—	—	—	—	—	—	○	—

## Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	n	Number of inputs	0 to 5	16-bit unsigned binary	—	User
+1	E1	Input value 1	-999999 to 999999	Single-precision real number	—	User
+2						
+3	E2	Input value 2				
+4						
⋮	⋮	⋮				
+2n-1	En	Input value n				
+2n						

## Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	BW	Output value	-999999 to 999999	Single-precision real number	—	System
+1						

## Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	n	Number of coefficients	0 to 5	16-bit unsigned binary	0	User
+1	K1	Coefficient 1	-999999 to 999999	Single-precision real number	1.0	User
+2						
+3	K2	Coefficient 2				
+4						
⋮	⋮	⋮				
+2n-1	Kn	Coefficient n				
+2n						
+2n+1	B	Bias	-999999 to 999999	Single-precision real number	0.0	User
+2n+2						

## Processing details

This instruction adds the data of input values (E1 to En) with a coefficient.

The S.ADD instruction performs the following operations.

$$BW=(K1 \times E1)+(K2 \times E2) \dots +(Kn \times En)+B$$

When the number of inputs (n) is 0, the output value (BW) becomes bias (B).

## Operation error

Error code	Error content
3402H	The value specified by (s1) or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The number of inputs (n) is less than 0 or greater than 5.
	The number of coefficients (n) is less than 0 or greater than 5.

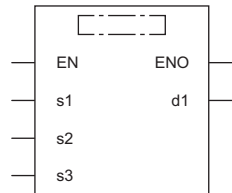
# Subtraction

## S.SUB

This instruction subtracts input data with a coefficient.

Ladder	ST
	<pre>ENO:=S_SUB(EN,s1,s2,s3,d1);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.SUB	

### Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Single-precision real number
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	—	—	—	—	—	—	—	—	—	○	—

## Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	n	Number of inputs	0 to 5	16-bit unsigned binary	—	User
+1	E1	Input value 1	-999999 to 999999	Single-precision real number	—	User
+2						
+3	E2	Input value 2				
+4						
⋮	⋮	⋮				
+2n-1	En	Input value n				
+2n						

## Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	BW	Output value	-999999 to 999999	Single-precision real number	—	System
+1						

## Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	n	Number of coefficients	0 to 5	16-bit unsigned binary	0	User
+1	K1	Coefficient 1	-999999 to 999999	Single-precision real number	1.0	User
+2						
+3	K2	Coefficient 2				
+4						
⋮	⋮	⋮				
+2n-1	Kn	Coefficient n				
+2n						
+2n+1	B	Bias	-999999 to 999999	Single-precision real number	0.0	User
+2n+2						

## Processing details

This instruction performs subtraction of the data of input values (E1 to En) with a coefficient.

The S.SUB instruction performs the following operations.

$$BW=(K1 \times E1)-(K2 \times E2) \dots -(Kn \times En)+B$$

When the number of inputs (n) is 0, the output value (BW) becomes bias (B).

## Operation error

Error code	Error content
3402H	The value specified by (s1) or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The number of inputs (n) is less than 0 or greater than 5.
	The number of coefficients (n) is less than 0 or greater than 5.

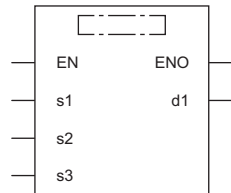
# Multiplication

## S.MUL

This instruction multiplies input data with a coefficient.

Ladder	ST
	<pre>ENO:=S_MUL(EN,s1,s2,s3,d1);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.MUL	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Single-precision real number
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

#### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	—	—	—	—	—	—	—	—	—	○	—

## Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	n	Number of inputs	0 to 5	16-bit unsigned binary	—	User
+1	E1	Input value 1	-999999 to 999999	Single-precision real number	—	User
+2						
+3	E2	Input value 2				
+4						
⋮	⋮	⋮				
+2n-1	En	Input value n				
+2n						

## Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	BW	Output value	-999999 to 999999	Single-precision real number	—	System
+1						

## Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	n	Number of coefficients	0 to 5	16-bit unsigned binary	0	User
+1	K1	Coefficient 1	-999999 to 999999	Single-precision real number	1.0	User
+2						
+3	K2	Coefficient 2				
+4						
⋮	⋮	⋮				
+2n-1	Kn	Coefficient n				
+2n						
+2n+1	B	Bias	-999999 to 999999	Single-precision real number	0.0	User
+2n+2						

## Processing details

This instruction performs multiplication of the data of input values (E1 to En) with a coefficient.

The S.MUL instruction performs the following operations.

$$BW=(K1 \times E1) \times (K2 \times E2) \dots \times (Kn \times En) + B$$

When the number of inputs (n) is 0, the output value (BW) becomes bias (B).

## Operation error

Error code	Error content
3402H	The value specified by (s1) or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	The number of inputs (n) is less than 0 or greater than 5.
	The number of coefficients (n) is less than 0 or greater than 5.

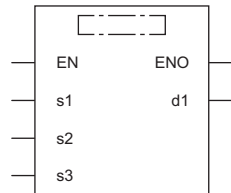
# Division

## S.DIV

This instruction performs division of the input data with a coefficient.

Ladder	ST
	ENO:=S_DIV(EN,s1,s2,s3,d1);

### FBD/LD



### Execution condition

Instruction	Execution condition
S.DIV	

### Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Single-precision real number
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

### Applicable devices

Operand	Bit		Word		Z	Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□□□, U3E□\□(H)G□		LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	—	—	—	—	—	—	—	—	—	○	—

### Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Input value 1 (numerator)	-999999 to 999999	Single-precision real number	—	User
+2 +3	E2	Input value 2 (denominator)	-999999 to 999999	Single-precision real number	—	User

## ■Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	-999999 to 999999	Single-precision real number	—	System

## ■Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	A	Coefficient 1	-999999 to 999999	Single-precision real number	1.0	User
+2 +3	K1	Coefficient 2	-999999 to 999999	Single-precision real number	1.0	User
+4 +5	K2	Coefficient 3	-999999 to 999999	Single-precision real number	1.0	User
+6 +7	B1	Bias 1	-999999 to 999999	Single-precision real number	0.0	User
+8 +9	B2	Bias 2	-999999 to 999999	Single-precision real number	0.0	User
+10 +11	B3	Bias 3	-999999 to 999999	Single-precision real number	0.0	User

## Processing details

This instruction divides the input value 1 (E1) by the input value 2 (E2).

The S.DIV instruction performs the following operations.

$$BW = A \times \frac{K1 \times E1 + B1}{K2 \times E2 + B2} + B3$$

When the denominator (efficient 2 (K2) × input value 2 (numerator) (E2) + bias 2 (B2)) is 0, the output value (BW) becomes bias 3 (B3).

## Operation error

Error code	Error content
3400H	An invalid operation (such as division by zero) is performed.
3402H	The value specified by (s1) or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.



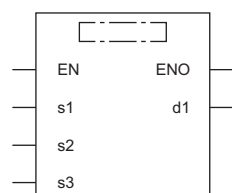
# Square root

## S.SQR

This instruction outputs the square root ( $\sqrt{\quad}$ ) of input data.

Ladder	ST
	<pre>ENO:=S_SQR(EN,s1,s2,s3,d1);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.SQR	

### Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Single-precision real number
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

### Applicable devices

Operand	Bit		Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□□G□, J□□□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	○	—	—	—	—	
(s3)	—	—	—	—	—	—	—	—	—	○	—	

### Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Input value	0 to 999999	Single-precision real number	—	User

## ■Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	0 to 999999	Single-precision real number	—	System

## ■Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	OLC	Output low-cut value	0 to 999999	Single-precision real number	0.0	User
+2 +3	K	Coefficient	0 to 999999	Single-precision real number	1.0	User

## Processing details

This instruction outputs  $\sqrt{\phantom{x}}$  of the input value (E1). When the input value (E1) is less than 0, 0 is output.

The S.SQR instruction performs the following operations.

$$BW = K \times \sqrt{E1}$$

In the following case, however, the output value (BW) becomes 0.

$$\text{Coefficient (K)} \times \sqrt{\text{Input value (E1)}} \leq \text{Output low-cut value (OLC)}$$

## Operation error

Error code	Error content
3402H	The value specified by (s1) or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

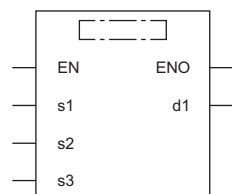
# Absolute value

## S.ABS

This instruction outputs the absolute value of input data.

Ladder	ST
	<pre>ENO:=S_ABS(EN,s1,s2,s3,d1);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.ABS	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Single-precision real number
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Empty string specification (Specify the empty string by enclosing it in double quotes (") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes (") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

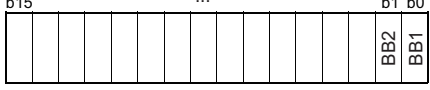
#### Applicable devices

Operand	Bit		Word			Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□□□, J□□□, U3E□(H)□□	Z	LT, LST, LC	LZ	K, H		E	\$		
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	—	○	—	—	—	—	
(s2)	—	—	—	—	—	—	—	—	—	—	○	—	
(s3)	—	—	—	—	—	—	—	—	—	—	○	—	

#### Input data

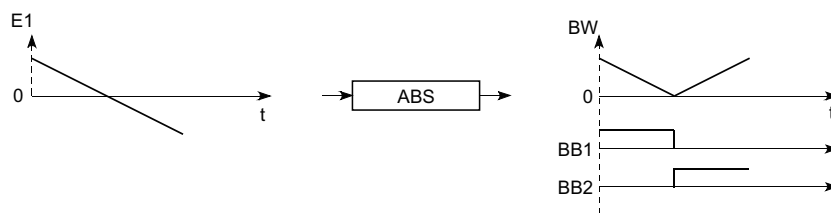
Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value	-999999 to 999999	Single-precision real number	—	User
+1						

## ■Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	0 to 999999	Single-precision real number	—	System
+2	BB	Sign determination of Input value (E1)	 <p>E1 &gt; 0: BB1 = 1 E1 &lt; 0: BB2 = 1 E1 = 0: BB1 = BB2 = 0</p>	16-bit unsigned binary	—	System

### Processing details

This instruction outputs the absolute value of the input value (E1).



The S.ABS instruction performs the following operations.

$$BW = |E1|$$

The instruction determines the sign of the input value (E1) and outputs the result to the sign determination bits (BB1 and BB2) of the input value (E1).

E1 status	BB1	BB2
E1 > 0	1	0
E1 < 0	0	1
E1 = 0	0	0

### Operation error

Error code	Error content
3402H	(s1) is a subnormal number or NaN (not a number).

# 21.7 Comparison Operation Instructions

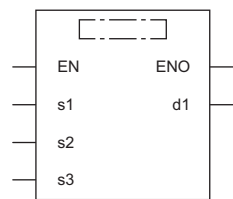
## Comparing data

### S.>

This instruction compares input data, and outputs the comparison result.

Ladder	ST
	ENO:=S_GT(EN,s1,s2,s3,d1);

### FBD/LD



(□ is to be replaced by S\_GT.)

### Execution condition

Instruction	Execution condition
S.>	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes (") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit


#### Applicable devices

Operand	Bit		Word		Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□\□, U3E□\(\H)G□	Z	LT, LST, LC		LZ	K	H	
(s1)	—	—	○	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—
(s3)	—	—	—	—	—	—	—	—	—	○	—

## Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Input value 1	-999999 to 999999	Single-precision real number	—	User
+2 +3	E2	Input value 2	-999999 to 999999	Single-precision real number	—	User

## Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	The same value as the input value 1 (E1) is stored.	Single-precision real number	—	System
+2	BB	BB1	Comparison output	16-bit unsigned binary	—	System
						
			The comparison result of E1 and E2 is stored.			

## Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	K	Set value	-999999 to 999999	Single-precision real number	0.0	User
+2 +3	HS	Hysteresis	0 to 999999	Single-precision real number	0.0	User

## Processing details

This instruction compares input values 1 (E1) and 2 (E2) and outputs the result (d1) to the comparison output (BB1).

Condition	BB1
$E1 > (E2 + K)$	1
$E1 \leq (E2 + K - HS)$	0
$(E2 + K - HS) < E1 \leq (E2 + K)$	Last value is output.

## Operation error

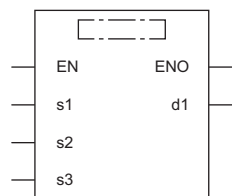
Error code	Error content
3402H	The value specified by (s1) or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Hysteresis (HS) is less than 0.

## S.<

This instruction compares input data, and outputs the comparison result.

Ladder	ST
	<pre>ENO:=S_LT(EN,s1,s2,s3,d1);</pre>

### FBD/LD



(□ is to be replaced by S\_LT.)

### Execution condition

Instruction	Execution condition
S.<	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

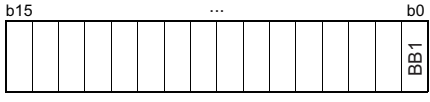
#### Applicable devices

Operand	Bit		Word		Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□\□, U3E□(H)G□	Z	LT, LST, LC		LZ	K, H	E	
(s1)	—	—	○	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—
(s3)	—	—	—	—	—	—	—	—	—	○	—

#### Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value 1	-999999 to 999999	Single-precision real number	—	User
+1						
+2	E2	Input value 2	-999999 to 999999	Single-precision real number	—	User
+3						

## ■Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	The same value as the input value 1 (E1) is stored.	Single-precision real number	—	System
+2	BB BB1	Comparison output	 <p>The comparison result of E1 and E2 is stored.</p>	16-bit unsigned binary	—	System

## ■Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	K	Set value	-999999 to 999999	Single-precision real number	0.0	User
+2 +3	HS	Hysteresis	0 to 999999	Single-precision real number	0.0	User

## Processing details

This instruction compares input values 1 (E1) and 2 (E2) and outputs the result (d1) to the comparison output (BB1).

Condition	BB1
$E1 < (E2 + K)$	1
$E1 \geq (E2 + K + HS)$	0
$(E2 + K) \leq E1 < (E2 + K + HS)$	Last value is output.

## Operation error

Error code	Error content
3402H	The value specified by (s1) or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Hysteresis (HS) is less than 0.

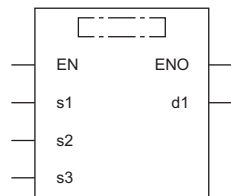


## S.=

This instruction compares input data, and outputs the comparison result.

Ladder	ST
	ENO:=S_EQ(EN,s1,s2,s3,d1);

### FBD/LD



(□ is to be replaced by S\_EQ.)

### Execution condition

Instruction	Execution condition
S.=	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Single-precision real number
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit


#### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□\□, U3E□\□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	—	—	—	—	—	—	—	—	—	○	—

#### Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value 1	-999999 to 999999	Single-precision real number	—	User
+1	E2	Input value 2	-999999 to 999999	Single-precision real number	—	User

## ■Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	The same value as the input value 1 (E1) is stored.	Single-precision real number	—	System
+2	BB BB1	Comparison output	 <p>The comparison result of E1 and E2 is stored.</p>	16-bit unsigned binary	—	System

## ■Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	K	Set value	-999999 to 999999	Single-precision real number	0.0	User

## Processing details

This instruction compares input values 1 (E1) and 2 (E2) and outputs the result (d1) to the comparison output (BB1).

Condition	BB1
$E1=(E2+K)$	1
$E1\neq(E2+K)$	0

## Operation error

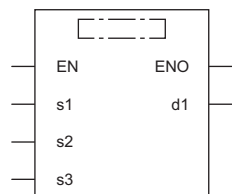
Error code	Error content
3402H	The value specified by (s1) or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.

## S.>=

This instruction compares input data, and outputs the comparison result.

Ladder	ST
	ENO:=S_GE(EN,s1,s2,s3,d1);

### FBD/LD



(□ is to be replaced by S\_GE.)

### Execution condition

Instruction	Execution condition
S.>=	

### Setting data

### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes ("") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

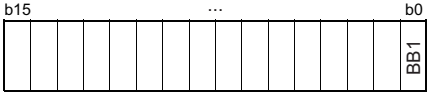
### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□\□, U3E□\□\G□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(s3)	—	—	—	—	—	—	—	—	—	—	○	—

### Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value 1	-999999 to 999999	Single-precision real number	—	User
+1						
+2	E2	Input value 2	-999999 to 999999	Single-precision real number	—	User
+3						

## ■Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	The same value as the input value 1 (E1) is stored.	Single-precision real number	—	System
+2	BB BB1	Comparison output	 <p>The comparison result of E1 and E2 is stored.</p>	16-bit unsigned binary	—	System

## ■Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	K	Set value	-999999 to 999999	Single-precision real number	0.0	User
+2 +3	HS	Hysteresis	0 to 999999	Single-precision real number	0.0	User

## Processing details

This instruction compares input values 1 (E1) and 2 (E2) and outputs the result (d1) to the comparison output (BB1).

Condition	BB1
$E1 \geq (E2 + K)$	1
$E1 < (E2 + K - HS)$	0
$(E2 + K - HS) \leq E1 < (E2 + K)$	Last value is output.

## Operation error

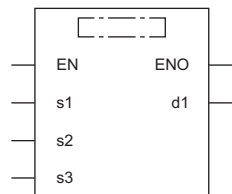
Error code	Error content
3402H	The value specified by (s1) or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Hysteresis (HS) is less than 0.

## S.<=

This instruction compares input data, and outputs the comparison result.

Ladder	ST
	ENO:=S_LE(EN,s1,s2,s3,d1);

### FBD/LD



(□ is to be replaced by S\_LE.)

### Execution condition

Instruction	Execution condition
S.<=	

### Setting data

#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	Word
(s2)	Operation constant start device	Refer to "Operation constant".	Word
(s3)	Empty string specification (Specify the empty string by enclosing it in double quotes (") for ladder. For ST and FBD/LD, use single quotes (').)	—	String
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit


#### Applicable devices

Operand	Bit		Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□\□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H, E, \$			
(s1)	—	—	○	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	○	—	—	—	—	
(s3)	—	—	—	—	—	—	—	—	—	○	—	

#### Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	E1	Input value 1	-999999 to 999999	Single-precision real number	—	User
+1						
+2	E2	Input value 2	-999999 to 999999	Single-precision real number	—	User
+3						

## ■Block memory

Operand: (d1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	BW	Output value	The same value as the input value 1 (E1) is stored.	Single-precision real number	—	System
+2	BB BB1	Comparison output	 <p>The comparison result of E1 and E2 is stored.</p>	16-bit unsigned binary	—	System

## ■Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	K	Set value	-999999 to 999999	Single-precision real number	0.0	User
+2 +3	HS	Hysteresis	0 to 999999	Single-precision real number	0.0	User

## Processing details

This instruction compares input values 1 (E1) and 2 (E2) and outputs the result (d1) to the comparison output (BB1).

Condition	BB1
$E1 \leq (E2 + K)$	1
$E1 > (E2 + K + HS)$	0
$(E2 + K) < E1 \leq (E2 + K + HS)$	Last value is output.

## Operation error

Error code	Error content
3402H	The value specified by (s1) or (s2) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	Hysteresis (HS) is less than 0.

# 21.8 Auto Tuning

Auto tuning is designed to make initial setting of PID constants.

Auto tuning can be used for processes that can be approximated with the "primary lag plus dead time" represented by the following expression.

**Ex.**

Process with relatively slow response such as temperature control

$$\frac{K}{1+Ts} e^{-Ls}$$

K: Gain, T: Time constant, L: Dead time, s: Laplace operator

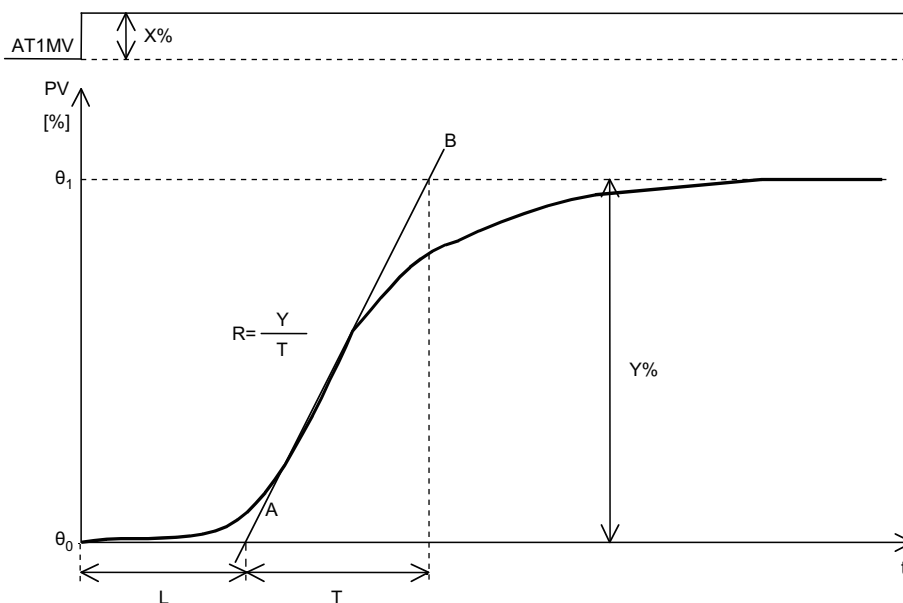
Auto tuning can be used for the loop that uses S.PID or S.2PID instruction.

Auto tuning is performed in the ZN method: step response method of Ziegler and Nichols.

## Outline of step response method

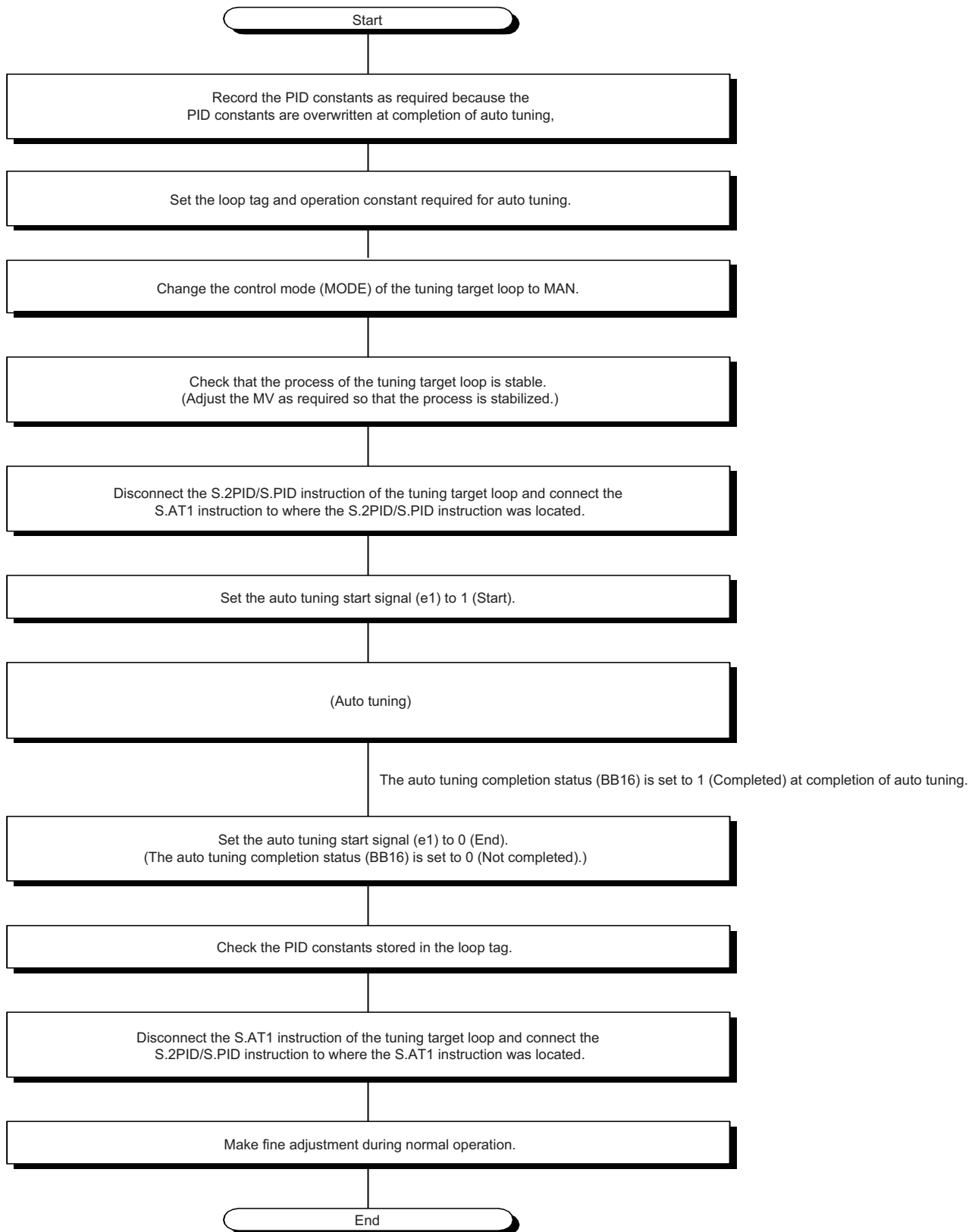
With no control operation being performed, change the manipulated value (MV) in a stepwise manner and look how the process value (PV) varies.

- When MV is changed in a stepwise manner, PV begins to change slowly. Soon, the change speed becomes faster and then becomes slow again, and finally is settled as a fixed value.
- Draw a tangent line at the place where PV varies fastest, and find the points of intersection A and B where the tangent line crosses the horizontal axis corresponding to the first process value ( $\theta_0$ ) and last process value ( $\theta_1$ ). This obtains the equivalent dead time (L) and equivalent time constant (T) as shown below.
- Determine maximum ramp (response speed)  $R = Y/T$  from the equivalent time constant (T) and maximum process value width (Y). Apply the equivalent dead time (L) and maximum ramp (R) to the Ziegler and Nichols' adjustment rule to calculate the proportional gain KP(P), integral constant TI(I), and derivative constant TD(D).



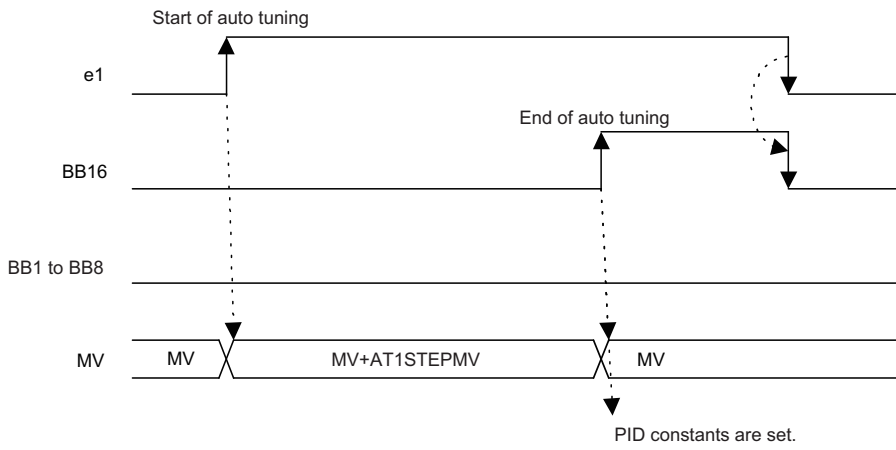
## Auto tuning procedure

The following shows the auto tuning procedure.

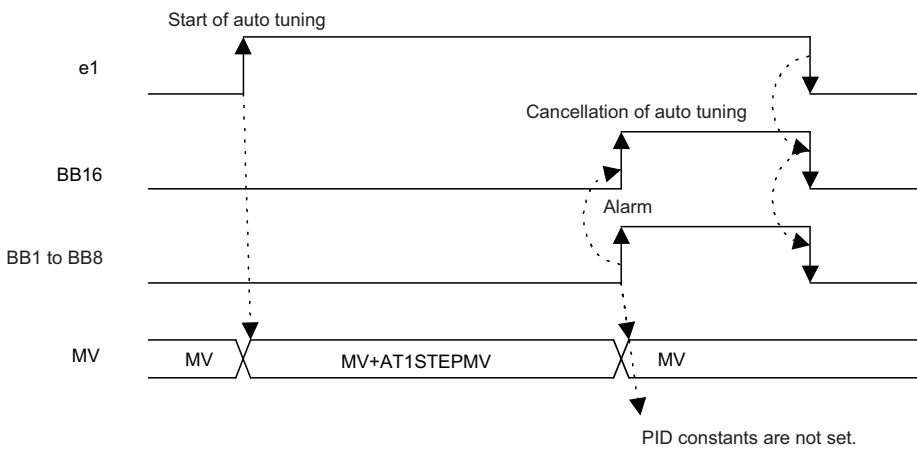




- Time chart from auto tuning start till normal completion



- Time chart from auto tuning start till stop due to alarm occurrence



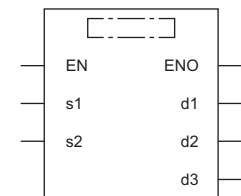
# Auto tuning instructions

## S.AT1

This instruction performs auto tuning to make initial setting of PID constants.

Ladder	ST
	<pre>ENO:=S_AT1(EN,s1,s2,d1,d2,d3);</pre>

### FBD/LD



### Execution condition

Instruction	Execution condition
S.AT1	

### Setting data


#### Description, range, data type

Operand	Description	Range	Data type
(s1)	Input data start device	Refer to "Input data".	Word
(d1)	Block memory start device	Refer to "Block memory".	16-bit unsigned binary
(s2)	Operation constant start device	Refer to "Operation constant".	16-bit unsigned binary
(d2)	Loop tag memory start device	Refer to "Loop tag memory".	Word
(d3)	Local work memory start device	Refer to "Local work memory".	Word
EN	Execution condition	—	Bit
ENO	Execution result	—	Bit

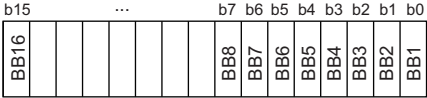
#### Applicable devices

Operand	Bit		Word		Double word			Indirect specification	Constant				Others
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, R, ZR, RD	U□\G□, J□\□, U3E□(H)G□	Z	LT, LST, LC	LZ		K, H	E	\$		
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	
(d1)	—	—	○	—	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	—	○	○	—	—	—	
(d2)	—	—	○	—	—	—	—	○	—	—	—	—	
(d3)	—	—	○	—	—	—	—	○	—	—	—	—	

## Input data

Operand: (s1)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0 +1	E1	Input value	-999999 to 999999 [%]	Single-precision real number	—	User
+2	e1	Auto tuning start signal	 <p>0: Stop/End 1: Start</p>	16-bit unsigned binary	0	User

## Block memory

Operand: (d1)																								
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by																		
+0	BB	<table border="1"> <tr> <td>BB1</td> <td>Alarm</td> </tr> <tr> <td>BB2</td> <td>Input upper limit alarm</td> </tr> <tr> <td>BB3</td> <td>Input lower limit alarm</td> </tr> <tr> <td>BB4</td> <td>Output upper limit alarm</td> </tr> <tr> <td>BB5</td> <td>Output lower limit alarm</td> </tr> <tr> <td>BB6</td> <td>Timeout alarm</td> </tr> <tr> <td>BB7</td> <td>Control mode alarm</td> </tr> <tr> <td>BB8</td> <td>Identification alarm</td> </tr> <tr> <td>BB16</td> <td>Auto tuning completion</td> </tr> </table>	BB1	Alarm	BB2	Input upper limit alarm	BB3	Input lower limit alarm	BB4	Output upper limit alarm	BB5	Output lower limit alarm	BB6	Timeout alarm	BB7	Control mode alarm	BB8	Identification alarm	BB16	Auto tuning completion	 <p>BB16 0: Not completed 1: Completed BB1 to BB8 0: No alarm 1: Alarm</p>	16-bit unsigned binary	—	System
BB1	Alarm																							
BB2	Input upper limit alarm																							
BB3	Input lower limit alarm																							
BB4	Output upper limit alarm																							
BB5	Output lower limit alarm																							
BB6	Timeout alarm																							
BB7	Control mode alarm																							
BB8	Identification alarm																							
BB16	Auto tuning completion																							

## Operation constant

Operand: (s2)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	PN	Operation mode	0: Reverse action 1: Direct action	16-bit unsigned binary	0	User

## Loop tag memory

The loop tag memory occupies 96 words from the specified start device.

Operand: (d2)																																			
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by																													
+1	MODE	Control mode	0 to FFFFH <table border="1" style="margin-left: 20px;"> <tr> <td>b15</td><td>...</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td></td><td></td><td></td><td></td><td></td><td>CSV</td><td>CMV</td><td>CCB</td><td>CAB</td><td>CMB</td><td>CAS</td><td>AUT</td><td>MAN</td><td>LOC</td><td>LCA</td><td>LCM</td> </tr> </table>	b15	...	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0						CSV	CMV	CCB	CAB	CMB	CAS	AUT	MAN	LOC	LCA	LCM	16-bit unsigned binary	8H	User/system
b15	...	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																							
					CSV	CMV	CCB	CAB	CMB	CAS	AUT	MAN	LOC	LCA	LCM																				
+3	ALM	Alarm detection	0 to FFFFH <table border="1" style="margin-left: 20px;"> <tr> <td>b15</td><td>b14</td><td>...</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>...</td><td>b0</td> </tr> <tr> <td></td><td>SPA</td><td></td><td></td><td></td><td></td><td>HHA</td><td>LLA</td><td>PHA</td><td>PLA</td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table> <p>SPA            0: Loop RUN            1: Loop STOP            HHA, LLA, PHA, PLA            0: No alarm            1: Alarm</p>	b15	b14	...	b8	b7	b6	b5	...	b0		SPA					HHA	LLA	PHA	PLA							16-bit unsigned binary	4000H	User/system				
b15	b14	...	b8	b7	b6	b5	...	b0																											
	SPA					HHA	LLA	PHA	PLA																										
+12 +13	MV	Manipulated value	-10 to 110 [%]	Single-precision real number	0.0	User/system																													
+18 +19	MH	Output upper limit value	-10 to 110 [%]	Single-precision real number	100.0	User																													
+20 +21	ML	Output lower limit value	-10 to 110 [%]	Single-precision real number	0.0	User																													
+52 +53	P	Gain	0 to 999999	Single-precision real number	1.0	User/system																													
+54 +55	I	Integral constant	0 to 999999 [s]	Single-precision real number	10.0	User/system																													
+56 +57	D	Derivative constant	0 to 999999 [s]	Single-precision real number	0.0	User/system																													
+70 +71	AT1STEPMV	AT1 step manipulated value	-100 to 100 [%]	Single-precision real number	0.0	User																													
+72 +73	AT1ST	AT1 sampling cycle	0 to 999999 [s] Set a value within the following range. $\frac{AT1ST}{\Delta T} \leq 32767$	Single-precision real number	1.0	User																													
+74 +75	AT1TOUT1	AT1 timeout time	0 to 999999 [s] Set a value within the following range. $\frac{AT1TOUT1}{\Delta T} \leq 32767$	Single-precision real number	100.0	User																													
+76 +77	AT1TOUT2	Timeout time after maximum AT1 ramp	0 to 999999 [s] Set a value within the following range. $\frac{AT1TOUT2}{\Delta T} \leq 32767$	Single-precision real number	10.0	User																													

## Local work memory

The system uses this area as a work area.

To start the control from the initial status, clear data by using a sequence program.

Operand: (d3)						
Device	Symbol	Name	Recommended range	Data type	Standard value	Set by
+0	—	Sampling period counter initialization completion flag	—	—	—	System
+1		Sampling period counter <sup>*1</sup>				
+2		Timeout time counter initialization completion flag				
+3		Timeout time counter <sup>*1</sup>				
+4		Timeout time (after maximum ramp) counter initialization completion flag				
+5		Timeout time (after maximum ramp) counter <sup>*1</sup>				
+6		Step manipulated value set completion flag				
+7		Auto tuning counter				
+8		Auto tuning start PV0				
+9						
+10	PV <sub>n-1</sub>	Last process value				
+11						
+12	—	Maximum ramp value				
+13						
+14		Counter at maximum ramp				
+15						
+16		PV at maximum ramp				
+17						
+18	R	Response speed				
+19						
+20	L	Equivalent dead time				
+21						

\*1 The sampling period counter, timeout time counter, and timeout (after maximum ramp) counter are each rounded off to the nearest whole number.

## Execution cycle ( $\Delta T$ )

Set the execution cycle ( $\Delta T$ ) in SD816 and SD817 in single-precision real number.

## Processing details

This instruction performs auto tuning to make initial setting of PID constants.

### ■Start signal determination processing

The instruction performs the following processing according to the auto tuning start signal (e1) and auto tuning completion (BB16).

e1	BB16	Processing
0	0	<ul style="list-style-type: none"> <li>• BB alarms from BB1 to identification alarm BB8 are set to 0.</li> <li>• When the step manipulated value set completion flag is 1, the following processing is performed. MV=MV-AT1STEPMV</li> <li>• The S.AT1 instruction is terminated.</li> </ul>
1	0	<ul style="list-style-type: none"> <li>• "Loop stop processing" is performed.</li> </ul>
0	1	<ul style="list-style-type: none"> <li>• BB auto tuning completion (BB16) is set to 0.</li> <li>• The S.AT1 instruction is terminated.</li> </ul>
1	1	<ul style="list-style-type: none"> <li>• The S.AT1 instruction is terminated.</li> </ul>

### ■Loop stop processing

The following processing is performed according to the SPA status of the alarm detection (ALM).

SPA status	Processing details
1	<p>The loop stops. When the loop stops, the following operations are performed and the S.AT1 instruction ends.</p> <ul style="list-style-type: none"> <li>• Auto tuning completion (BB16) is set to 1.</li> <li>• When the step manipulated value set completion flag is 1, the following processing is performed. MV=MV-AT1STEPMV</li> </ul>
0	The loop runs, and "mode determination processing" is performed.

### ■Mode determination processing

The following processing is performed depending on the control mode (MODE) setting.

Control mode (MODE) setting	Processing details
AUT, CAB, CAS, CCB, CSV, LCA, LCC	<p>The following processing is performed and the S.AT1 instruction is terminated.</p> <ul style="list-style-type: none"> <li>• The control mode alarm (BB7) is set to 1.</li> <li>• Auto tuning completion (BB16) is set to 1.</li> <li>• When the step manipulated value set completion flag is 1, the following processing is performed. MV=MV-AT1STEPMV</li> </ul>
MAN, CMB, CMV, LCM	"Input check processing" is performed.

### ■Input check processing

The following processing is performed according to the alarm detection (ALM).

Alarm Detection (ALM)	Processing details
PHA or HHA is 1.	<p>The following processing is performed and the S.AT1 instruction is terminated.</p> <ul style="list-style-type: none"> <li>• The input upper limit alarm (BB2) is set to 1.</li> <li>• Auto tuning completion (BB16) is set to 1.</li> </ul>
PLA or LLA is 1.	<p>The following processing is performed and the S.AT1 instruction is terminated.</p> <ul style="list-style-type: none"> <li>• The input upper limit alarm (BB3) is set to 1.</li> <li>• Auto tuning completion (BB16) is set to 1.</li> </ul>

### ■Timeout determination processing

This function determines whether the auto tuning processing has reached the AT1 timeout time (AT1TOUT1).

Auto tuning processing	Processing details
If the AT1 timeout time (AT1TOUT1) is reached	<p>The following processing is performed and the S.AT1 instruction is terminated.</p> <ul style="list-style-type: none"> <li>• The input upper limit alarm (BB6) is set to 1.</li> <li>• Auto tuning completion (BB16) is set to 1.</li> </ul>
If the AT1 timeout time (AT1TOUT1) has not been reached	"Timeout (after maximum ramp) determination processing" is performed.

### ■Timeout (after maximum ramp) determination processing

This function determines whether the auto tuning processing has reached the AT1 timeout time after maximum ramp (AT1TOUT2).

However, if the timeout time (after maximum ramp) counter initialization completion flag is 0, "step manipulated value set processing" is performed.

Auto tuning processing	Processing details
If the AT1 timeout time after maximum ramp (AT1TOUT2) is reached	"Identification processing" is performed.
If the AT1 timeout time after maximum ramp (AT1TOUT2) has not been reached	"Step manipulated value set processing" is performed.

### ■Step manipulated value set processing

This function checks the step manipulated value set completion flag to determine whether the step manipulated value has been set (1) or not set (0).

- When the step manipulated value set completion flag is 0, the following processing is performed.

The AT1 step manipulated value (AT1SETPMV) is added to the manipulated value (MV).

$$T1 = MV + AT1STEPMV$$

The upper/lower limiter function performs the following operations, and outputs the result to the output upper limit alarm (BB4) and output lower limit alarm (BB5).

Condition	Result			MV	Processing after upper/lower limiter
	BB4	BB5	BB16		
$T1 > MH$	1	0	1	Original MV remains unchanged.	The S.AT1 instruction is terminated.
$T1 < ML$	0	1	1		
$ML \leq T1 \leq MH$	0	0	0	T1	The following processing is performed. <ul style="list-style-type: none"> <li>• The step manipulated value set completion flag is set to 1.</li> <li>• The auto tuning counter is set to 0.</li> <li>• The input value (E1) is stored in auto tuning start PV0.</li> <li>• The input value (E1) is stored in the last measurement value (PV<sub>n-1</sub>).</li> <li>• The maximum slope value, maximum ramp counter, maximum ramp PV, response speed (R), and equivalent dead time (L) are set to 0.</li> </ul>

- When the step manipulated value set completion flag is 1, "sampling cycle determination processing" is performed.

### ■Sampling cycle determination processing

This function checks the AT1 sampling period (AT1ST) to determine whether the sampling period has been reached.

- If the sampling period has not been reached, the S.AT1 instruction is terminated.
- If the sampling period has been reached, "response waveform observation processing" is performed.

## ■Response waveform observation processing

The following processing is performed for the input value (E1).

Item	Processing		
Response waveform observation	The auto tuning counter is incremented.		
	The following processing is performed according to the input value (E1) and last measurement value (PV <sub>n-1</sub> ). T2=E1-PV <sub>n-1</sub> <ul style="list-style-type: none"> <li>Reserve action (PN = 0)</li> <li>Direct action (PN = 1)</li> </ul>		
	The input value (E1) is stored in the last measurement value (PV <sub>n-1</sub> ).		
Maximum ramp value	The following processing is performed according to the ramp (T2) and the S.AT1 instruction is terminated.		
	<ul style="list-style-type: none"> <li>Reverse action (PN=0) and AT1 step manipulated value (AT1STEP MV) ≥ 0</li> <li>Direct action (PN=1) and AT1 step manipulated value (AT1STEP MV) &lt; 0</li> </ul>	Maximum ramp value ≤ Ramp (T2)	<ul style="list-style-type: none"> <li>Maximum ramp value = Ramp (T2)</li> <li>Maximum ramp counter = Counter from the start of auto tuning</li> <li>Maximum ramp PV = Input value (E1)</li> <li>Reset the timeout time (after maximum ramp) counter, and start counting again.</li> </ul>
		Maximum ramp value > Ramp (T2)	The last maximum ramp value remains unchanged.
	<ul style="list-style-type: none"> <li>Direct action (PN=1) and AT1 step manipulated value (AT1STEP MV) ≥ 0</li> <li>Reverse action (PN=0) and AT1 step manipulated value (AT1STEP MV) &lt; 0</li> </ul>	Maximum ramp value ≥ Ramp (T2)	<ul style="list-style-type: none"> <li>Maximum ramp value = Ramp (T2)</li> <li>Maximum ramp counter = Counter from the start of auto tuning</li> <li>Maximum ramp PV = Input value (E1)</li> <li>Reset the timeout time (after maximum ramp) counter, and start counting again.</li> </ul>
	Maximum ramp value < Ramp (T2)	The last maximum ramp value remains unchanged.	

## ■Identification processing

The following processing is performed from the maximum ramp value.

Processing	Item
Response speed	The response speed for calculation (R') and response speed (R) are calculated from the following expressions. $R' = \frac{\text{Maximum ramp value (\%)}}{\text{AT1ST}}, R = \frac{ R' }{100} \text{ (/s)}$
	When the response speed (R) is 0, the following processing is performed and the S.AT1 instruction is terminated. <ul style="list-style-type: none"> <li>Identification alarm BB8 is set to 1.</li> <li>Auto tuning completion (BB16) is set to 1.</li> <li>When the step manipulated value set completion flag is 1, the following processing is performed. MV=MV-AT1STEP MV</li> </ul>
Equivalent dead time	The Y-axis intercept (b) when tangent is drawn by the response speed for calculation (R') and the equivalent dead time (L) are calculated from the following expressions. $b = (\text{PV at maximum ramp}) - R' \times (\text{counter at maximum ramp}) \times \text{AT1ST}$ $L = \frac{(\text{Auto tuning start-time PV0}) - b}{R'}$
	When the equivalent dead time (L) is equal to or less than 0, the following processing is performed and the S.AT1 instruction is terminated. <ul style="list-style-type: none"> <li>Identification alarm BB8 is set to 1.</li> <li>Auto tuning completion (BB16) is set to 1.</li> <li>When the step manipulated value set completion flag is 1, the following processing is performed. MV=MV-AT1STEP MV</li> </ul>



## ■PID constants calculation processing

The response speed (R), equivalent dead time (L), and AT1 step manipulated value (AT1STEPMV) are assigned to the adjustment rule to calculate PID constants.

- Control method

The control method is selected according to the integral constant TI (I) and derivative constant TD (D).

Integral constant $T_I(I)$	Derivative constant $T_D(D)$	Control method
$T_I \leq 0$	—	Proportional control (P action) only
$T_I > 0$	$T_D \leq 0$	PI control (PI action)
	$T_D > 0$	PID control (PID action)

- Adjustment rule

N method: The adjustment rule using the step response of Ziegler and Nichols is used.

Control method	Proportional gain $K_P(P)$	Integral constant $T_I(I)$	Derivative constant $T_D(D)$
P	$\frac{1}{R \times L} \times \frac{ AT1STEPMV }{100}$	0	0
PI	$\frac{0.9}{R \times L} \times \frac{ AT1STEPMV }{100}$	$3.33 \times L$	0
PID	$\frac{1.2}{R \times L} \times \frac{ AT1STEPMV }{100}$	$2 \times L$	$0.5 \times L$

- PID constants storing

The following processing is performed and the S.AT1 instruction is terminated.

- PID constants are stored in the gain (P), integral constants (I), and derivative constants (D).
- Auto tuning completion (BB16) is set to 1.
- The AT1 step manipulated value (AT1STEPMV) is subtracted from the manipulated value (MV) and the result is stored in the manipulated value (MV).  
 $MV = MV - AT1STEPMV$

## Operation error

Error code	Error content
3400H	An invalid operation (such as division by zero) is performed.
3402H	(s1) is a subnormal number or NaN (not a number).
3403H	An overflow has occurred.
3405H	AT1 sampling period (AT1ST) < 0
	AT1 timeout time (AT1TOUT1) < 0
	AT1 timeout (after maximum ramp) time (AT1TOUT2) < 0
	Execution cycle ( $\Delta T$ ) < 0
	(AT1 sampling period (AT1ST) ÷ execution cycle ( $\Delta T$ )) > 32767
(AT1 timeout time (AT1TOUT1) ÷ execution cycle ( $\Delta T$ )) > 32767	
(AT1 timeout (after maximum ramp) time (AT1TOUT2) ÷ Execution cycle ( $\Delta T$ )) > 32767	

## 21.9 Lists of Loop Tag Memory

### PID control (SPID), two-degree-of-freedom PID control (S2PID), sample PI control (SSPI)

Offset	Instruction used	Item	Name	Recommended range	Data type	Set by	
						SPID, S2PID	SSPI
+1	Shared among the instructions	MODE	Control mode	0 to FFFFH	16-bit unsigned binary	User/system	User/system
+3		ALM	Alarm detection	0 to FFFFH	16-bit unsigned binary	User/system	User/system
+4		INH	Disable alarm detection	0 to FFFFH	16-bit unsigned binary	User/system	User/system
+9	S.PHPL2	N	Number of digits after the decimal point	0 to 4	16-bit unsigned binary	User	User
+10	S.PHPL/ S.PHPL2	PV	Process value	RL to RH	Single-precision real number	System	System
+11							
+12	S.OUT1/ S.DUTY	MV	Manipulated value	-10 to 110 [%]	Single-precision real number	User/system	User/system
+13							
+14	S.PID/ S.2PID/S.SPI	SV	Set value	RL to RH	Single-precision real number	User	User
+15							
+16	S.PID/ S.2PID/S.SPI	DV	Deviation	-110 to 110 [%]	Single-precision real number	System	System
+17							
+18	S.OUT1/ S.DUTY	MH	Output upper limit value	-10 to 110 [%]	Single-precision real number	User	User
+19							
+20	S.OUT1/ S.DUTY	ML	Output lower limit value	-10 to 110 [%]	Single-precision real number	User	User
+21							
+22	S.PHPL/ S.PID/ S.2PID/S.SPI	RH	Engineering value upper limit	-999999 to 999999	Single-precision real number	User	User
+23							
+24	S.PHPL/ S.PID/ S.2PID/S.SPI	RL	Engineering value lower limit	-999999 to 999999	Single-precision real number	User	User
+25							
+26	S.PHPL/ S.PHPL2	PH	Upper limit alarm value	RL to RH PL < PH	Single-precision real number	User	User
+27							
+28	S.PHPL/ S.PHPL2	PL	Lower limit alarm value	RL to RH PL < PH	Single-precision real number	User	User
+29							
+30	S.PHPL/ S.PHPL2	HH	Upper upper limit alarm value	RL to RH PH ≤ HH	Single-precision real number	User	User
+31							
+32	S.PHPL/ S.PHPL2	LL	Lower lower limit alarm value	RL to RH LL ≤ PL	Single-precision real number	User	User
+33							
+38	S.IN	α	Filter coefficient	0 to 1	Single-precision real number	User	User
+39							
+40	S.PHPL/ S.PHPL2	HS	Upper/lower limit alarm hysteresis	0 to 999999 [%]	Single-precision real number	User	User
+41							
+42	S.PHPL/ S.PHPL2	CTIM	Variation rate alarm check time	0 to 999999 [s]	Single-precision real number	User	User
+43							
+44	S.PHPL/ S.PHPL2	DPL	Variation rate alarm value	0 to 100 [%]	Single-precision real number	User	User
+45							
+46	S.PID/ S.2PID/S.SPI	CT/ST	Control cycle/Operating time	0 to 999999 [s]	Single-precision real number	User (Set CT.)	User (Set ST.)
+47							
+48	S.OUT1/ S.DUTY	DML	Output variation rate limit value	0 to 100 [%]	Single-precision real number	User	User
+49							
+50	S.PID/ S.2PID/S.SPI	DVL	Deviation limit value	0 to 100 [%]	Single-precision real number	User	User
+51							
+52	S.PID/ S.2PID/S.SPI	P	Gain	0 to 999999	Single-precision real number	User	User
+53							

Offset	Instruction used	Item	Name	Recommended range	Data type	Set by	
						SPID, S2PID	SSPI
+54 +55	S.PID/ S.2PID/ S.SPI/ S.OUT1/ S.DUTY	I*1	Integral constant	0 to 999999 [s]	Single-precision real number	User	User
+56 +57	S.PID/ S.2PID/S.SPI	D/STHT	Derivative constant/ Sampling cycle	0 to 999999 [s]	Single-precision real number	User (Set D.)	User (Set STHT.)
+58 +59	S.PID/ S.2PID/S.SPI	GW	Gap width	0 to 100 [%]	Single-precision real number	User	User
+60 +61	S.PID/ S.2PID/S.SPI	GG	Gap gain	0 to 999999	Single-precision real number	User	User
+62 +63	S.PID/ S.2PID/ S.SPI/ S.OUT1/ S.DUTY	MVP	MV internal operation value	-999999 to 999999 [%]	Single-precision real number	System	System
+64 +65	S.2PID	$\alpha$	Two-degree-of-freedom parameter $\alpha$	0 to 1	Single-precision real number	User	—
+66 +67	S.2PID	$\beta$	Two-degree-of-freedom parameter $\beta$	0 to 1	Single-precision real number	User	—
+68 +69	S.DUTY	CTDUTY	Control output cycle	0 to 999999 [s]	Single-precision real number	User	—

\*1 The following instruction pairs use the same value in I.

- S.PID instruction and S.OUT1 instruction
- S.PID instruction and S.DUTY instruction
- S.2PID instruction and S.OUT1 instruction
- S.2PID instruction and S.DUTY instruction
- S.SPI instruction and S.OUT1 instruction

# I-PD control (SIPD), blend PI control (SBPI)

Offset	Instruction used	Item	Name	Recommended range	Data type	Set by	
						SIPD	SBPI
+1	Shared among the instructions	MODE	Control mode	0 to FFFFH	16-bit unsigned binary	User/system	User/system
+3		ALM	Alarm detection	0 to FFFFH	16-bit unsigned binary	User/system	User/system
+4		INH	Disable alarm detection	0 to FFFFH	16-bit unsigned binary	User/system	User/system
+9	S.PHPL2	N	Number of digits after the decimal point	0 to 4	16-bit unsigned binary	User	User
+10 +11	S.PHPL/ S.PHPL2	PV	Process value	RH to RL	Single-precision real number	System	System
+12 +13	S.OUT1	MV	Manipulated value	-10 to 110 [%]	Single-precision real number	User/system	User/system
+14 +15	S.IPD/S.BPI	SV	Set value	RL to RH	Single-precision real number	User	User
+16 +17	S.IPD/S.BPI	DV	Deviation	-110 to 110 [%]	Single-precision real number	System	System
+18 +19	S.OUT1	MH	Output upper limit value	-10 to 110 [%]	Single-precision real number	User	User
+20 +21	S.OUT1	ML	Output lower limit value	-10 to 110 [%]	Single-precision real number	User	User
+22 +23	S.PHPL/ S.IPD/S.BPI	RH	Engineering value upper limit	-999999 to 999999	Single-precision real number	User	User
+24 +25	S.PHPL/ S.IPD/S.BPI	RL	Engineering value lower limit	-999999 to 999999	Single-precision real number	User	User
+26 +27	S.PHPL/ S.PHPL2	PH	Upper limit alarm value	RL to RH PL < PH	Single-precision real number	User	User
+28 +29	S.PHPL/ S.PHPL2	PL	Lower limit alarm value	RL to RH PL < PH	Single-precision real number	User	User
+30 +31	S.PHPL/ S.PHPL2	HH	Upper upper limit alarm value	RL to RH PH ≤ HH	Single-precision real number	User	User
+32 +33	S.PHPL/ S.PHPL2	LL	Lower lower limit alarm value	RL to RH LL ≤ PL	Single-precision real number	User	User
+38 +39	S.IN	$\alpha$	Filter coefficient	0 to 1	Single-precision real number	User	User
+40 +41	S.PHPL/ S.PHPL2	HS	Upper/lower limit alarm hysteresis	0 to 999999 [%]	Single-precision real number	User	User
+42 +43	S.PHPL/ S.PHPL2	CTIM	Variation rate alarm check time	0 to 999999 [s]	Single-precision real number	User	User
+44 +45	S.PHPL/ S.PHPL2	DPL	Variation rate alarm value	0 to 100 [%]	Single-precision real number	User	User
+46 +47	S.IPD/S.BPI	CT	Control cycle	0 to 999999 [s]	Single-precision real number	User	User
+48 +49	S.OUT1	DML	Output variation rate limit value	0 to 100 [%]	Single-precision real number	User	User
+50 +51	S.IPD/S.BPI	DVL	Deviation limit value	0 to 100 [%]	Single-precision real number	User	User
+52 +53	S.IPD/S.BPI	P	Gain	0 to 999999	Single-precision real number	User	User
+54 +55	S.IPD/S.BPI/ S.OUT1	$I^{*1}$	Integral constant	0 to 999999 [s]	Single-precision real number	User	User
+56 +57	S.IPD/S.BPI	D/SDV	Derivative constant/DV cumulative total	D: 0 to 999999 [s] SDV: -999999 to 999999 [%]	Single-precision real number	User —	— System
+58 +59	S.IPD/S.BPI	GW	Gap width	0 to 100 [%]	Single-precision real number	User	User

Offset	Instruction used	Item	Name	Recommended range	Data type	Set by	
						SIPD	SBPI
+60 +61	S.IPD/S.BPI	GG	Gap gain	0 to 999999	Single-precision real number	User	User
+62 +63	S.IPD/ S.OUT1	MVP	MV internal operation value	-999999 to 999999 [%]	Single-precision real number	System	—

\*1 The instruction pairs, S.IPD instruction and S.OUT1 instruction, S.BPI instruction and S.OUT1 instruction, use the same value in I.

## Manual output (SMOUT), monitor (SMON)

Offset	Instruction used	Item	Name	Recommended range	Data type	Set by	
						SMOUT	SMON
+1	Shared among the instructions	MODE	Control mode	0 to FFFFH	16-bit unsigned binary	User/system	User/system
+3		ALM	Alarm detection	0 to FFFFH	16-bit unsigned binary	User/system	User/system
+4		INH	Disable alarm detection	0 to FFFFH	16-bit unsigned binary	—	User/system
+9	S.PHPL2	N	Number of digits after the decimal point	0 to 4	16-bit unsigned binary	—	User
+10 +11	S.PHPL/ S.PHPL2	PV	Process value	RL to RH	Single-precision real number	—	System
+12 +13	S.MOUT	MV	Manipulated value	-10 to 110 [%]	Single-precision real number	User	—
+22 +23	S.PHPL	RH	Engineering value upper limit	-999999 to 999999	Single-precision real number	—	User
+24 +25	S.PHPL	RL	Engineering value lower limit	-999999 to 999999	Single-precision real number	—	User
+26 +27	S.PHPL/ S.PHPL2	PH	Upper limit alarm value	RL to RH PL < PH	Single-precision real number	—	User
+28 +29	S.PHPL/ S.PHPL2	PL	Lower limit alarm value	RL to RH PL < PH	Single-precision real number	—	User
+30 +31	S.PHPL/ S.PHPL2	HH	Upper upper limit alarm value	RL to RH PH ≤ HH	Single-precision real number	—	User
+32 +33	S.PHPL/ S.PHPL2	LL	Lower lower limit alarm value	RL to RH LL ≤ PL	Single-precision real number	—	User
+38 +39	S.IN	α	Filter coefficient	0 to 1	Single-precision real number	—	User
+40 +41	S.PHPL/ S.PHPL2	HS	Upper/lower limit alarm hysteresis	0 to 999999 [%]	Single-precision real number	—	User
+42 +43	S.PHPL/ S.PHPL2	CTIM	Variation rate alarm check time	0 to 999999 [s]	Single-precision real number	—	User
+44 +45	S.PHPL/ S.PHPL2	DPL	Variation rate alarm value	0 to 100 [%]	Single-precision real number	—	User

# Manual output with monitor (SMWM), PIDP control (SPIDP)

Offset	Instruction used	Item	Name	Recommended range	Data type	Set by	
						SMWM	SPIDP
+1	Shared among the instructions	MODE	Control mode	0 to FFFFH	16-bit unsigned binary	User/system	User/system
+3		ALM	Alarm detection	0 to FFFFH	16-bit unsigned binary	User/system	User/system
+4		INH	Disable alarm detection	0 to FFFFH	16-bit unsigned binary	User/system	User/system
+9	S.PHPL2	N	Number of digits after the decimal point	0 to 4	16-bit unsigned binary	User	User
+10	S.PHPL/ S.PHPL2	PV	Process value	RL to RH	Single-precision real number	System	System
+11		MV	Manipulated value	-10 to 110 [%]	Single-precision real number	User	User/system
+12	S.MOUT/ S.PIDP	SV	Set value	RL to RH	Single-precision real number	—	User
+13							
+14	S.PIDP	DV	Deviation	-110 to 110 [%]	Single-precision real number	—	System
+15							
+16	S.PIDP	MH	Output upper limit value	-10 to 110 [%]	Single-precision real number	—	User
+17							
+18	S.PIDP	ML	Output lower limit value	-10 to 110 [%]	Single-precision real number	—	User
+19							
+20	S.PIDP	RH	Engineering value upper limit	-999999 to 999999	Single-precision real number	User	User
+21							
+22	S.PHPL/ S.PIDP	RL	Engineering value lower limit	-999999 to 999999	Single-precision real number	User	User
+23							
+24	S.PHPL/ S.PHPL2	PH	Upper limit alarm value	RL to RH PL < PH	Single-precision real number	User	User
+25							
+26	S.PHPL/ S.PHPL2	PL	Lower limit alarm value	RL to RH PL < PH	Single-precision real number	User	User
+27							
+28	S.PHPL/ S.PHPL2	HH	Upper upper limit alarm value	RL to RH PH ≤ HH	Single-precision real number	User	User
+29							
+30	S.PHPL/ S.PHPL2	LL	Lower lower limit alarm value	RL to RH LL ≤ PL	Single-precision real number	User	User
+31							
+32	S.IN	α	Filter coefficient	0 to 1	Single-precision real number	User	User
+33							
+38	S.PHPL/ S.PHPL2	HS	Upper/lower limit alarm hysteresis	0 to 999999 [%]	Single-precision real number	User	User
+39							
+40	S.PHPL/ S.PHPL2	CTIM	Variation rate alarm check time	0 to 999999 [s]	Single-precision real number	User	User
+41							
+42	S.PHPL/ S.PHPL2	DPL	Variation rate alarm value	0 to 100 [%]	Single-precision real number	User	User
+43							
+44	S.PIDP	CT	Control cycle	0 to 999999 [s]	Single-precision real number	—	User
+45							
+46	S.PIDP	DML	Output variation rate limit value	0 to 100 [%]	Single-precision real number	—	User
+47							
+48	S.PIDP	DVL	Deviation limit value	0 to 100 [%]	Single-precision real number	—	User
+49							
+50	S.PIDP	P	Gain	0 to 999999	Single-precision real number	—	User
+51							
+52	S.PIDP	I	Integral constant	0 to 999999 [s]	Single-precision real number	—	User
+53							
+54	S.PIDP	D	Derivative constant	0 to 999999 [s]	Single-precision real number	—	User
+55							
+56	S.PIDP	GW	Gap width	0 to 100 [%]	Single-precision real number	—	User
+57							
+58	S.PIDP	GG	Gap gain	0 to 999999	Single-precision real number	—	User
+59							
+60	S.PIDP						
+61							

## Two-position (on/off) control (SONF2), three-position (on/off) control (SONF3)

Offset	Instruction used	Item	Name	Recommended range	Data type	Set by	
						SONF2	SONF3
+1	Shared among the instructions	MODE	Control mode	0 to FFFFH	16-bit unsigned binary	User/system	User/system
+3		ALM	Alarm detection	0 to FFFFH	16-bit unsigned binary	User/system	User/system
+4		INH	Disable alarm detection	0 to FFFFH	16-bit unsigned binary	User/system	User/system
+9	S.PHPL2	N	Number of digits after the decimal point	0 to 4	16-bit unsigned binary	User	User
+10	S.PHPL/ S.PHPL2	PV	Process value	RL to RH	Single-precision real number	System	System
+11							
+12	S.ONF2/ S.ONF3	MV	Manipulated value	-10 to 110 [%]	Single-precision real number	User/system	User/system
+13							
+14	S.ONF2/ S.ONF3	SV	Set value	RL to RH	Single-precision real number	User	User
+15							
+16	S.ONF2/ S.ONF3	DV	Deviation	-110 to 110 [%]	Single-precision real number	System	System
+17							
+18	S.ONF2/ S.ONF3	HSO	Hysteresis	0 to 999999	Single-precision real number	User	User
+19							
+20	S.ONF3	HS1	Hysteresis	0 to 999999	Single-precision real number	—	User
+21							
+22	S.PHPL	RH	Engineering value upper limit	-999999 to 999999	Single-precision real number	User	User
+23							
+24	S.PHPL	RL	Engineering value lower limit	-999999 to 999999	Single-precision real number	User	User
+25							
+26	S.PHPL/ S.PHPL2	PH	Upper limit alarm value	RL to RH PL < PH	Single-precision real number	User	User
+27							
+28	S.PHPL/ S.PHPL2	PL	Lower limit alarm value	RL to RH PL < PH	Single-precision real number	User	User
+29							
+30	S.PHPL/ S.PHPL2	HH	Upper upper limit alarm value	RL to RH PH ≤ HH	Single-precision real number	User	User
+31							
+32	S.PHPL/ S.PHPL2	LL	Lower lower limit alarm value	RL to RH LL ≤ PL	Single-precision real number	User	User
+33							
+38	S.IN	$\alpha$	Filter coefficient	0 to 1	Single-precision real number	User	User
+39							
+40	S.PHPL/ S.PHPL2	HS	Upper/lower limit alarm hysteresis	0 to 999999 [%]	Single-precision real number	User	User
+41							
+42	S.PHPL/ S.PHPL2	CTIM	Variation rate alarm check time	0 to 999999 [s]	Single-precision real number	User	User
+43							
+44	S.PHPL/ S.PHPL2	DPL	Variation rate alarm value	0 to 100 [%]	Single-precision real number	User	User
+45							
+46	S.ONF2/ S.ONF3	CT	Control cycle	0 to 999999 [s]	Single-precision real number	User	User
+47							



## Batch counter (SBC)

Offset	Instruction used	Item	Name	Recommended range	Data type	Set by
						SBC
+1	Shared among the instructions	MODE	Control mode	0 to FFFFH	16-bit unsigned binary	User/system
+3		ALM	Alarm detection	0 to FFFFH	16-bit unsigned binary	User/system
+4		INH	Disable alarm detection	0 to FFFFH	16-bit unsigned binary	User/system
+10 +11	S.PSUM	SUM1	Integrated value (integral part)	0 to 2147483647	32-bit unsigned binary	System
+12 +13	S.PSUM	SUM2	Integrated value (decimal part)	0 to 2147483647	32-bit unsigned binary	System
+14 +15	S.BC	SV1	Set value 1	0 to 2147483647	32-bit unsigned binary	User
+16 +17	S.BC	SV2	Set value 2	0 to 2147483647	32-bit unsigned binary	User
+26 +27	S.BC	PH	Upper limit alarm value	0 to 2147483647	32-bit unsigned binary	User
+42 +43	S.BC	CTIM	Variation rate alarm check time	0 to 999999 [s]	Single-precision real number	User
+44 +45	S.BC	DPL	Variation rate alarm value	0 to 2147483647	32-bit unsigned binary	User

## Ratio control (SR)

Offset	Instruction used	Item	Name	Recommended range	Data type	Set by
						SR
+1	Shared among the instructions	MODE	Control mode	0 to FFFFH	16-bit unsigned binary	User/system
+3		ALM	Alarm detection	0 to FFFFH	16-bit unsigned binary	User/system
+4		INH	Disable alarm detection	0 to FFFFH	16-bit unsigned binary	User/system
+9	S.PHPL2	N	Number of digits after the decimal point	0 to 4	16-bit unsigned binary	User
+10 +11	S.PHPL/ S.PHPL2	PV	Process value	RL to RH	Single-precision real number	System
+12 +13	S.OUT2	MV	Manipulated value	-10 to 110 [%]	Single-precision real number	User/system
+14 +15	S.R	SPR	Set value	-999999 to 999999	Single-precision real number	User
+16 +17	S.R	BIAS	Bias	-999999 to 999999 [%]	Single-precision real number	User
+18 +19	S.OUT2	MH	Output upper limit value	-10 to 110 [%]	Single-precision real number	User
+20 +21	S.OUT2	ML	Output lower limit value	-10 to 110 [%]	Single-precision real number	User
+22 +23	S.PHPL	RH	Engineering value upper limit	-999999 to 999999	Single-precision real number	User
+24 +25	S.PHPL	RL	Engineering value lower limit	-999999 to 999999	Single-precision real number	User
+26 +27	S.PHPL/ S.PHPL2	PH	Upper limit alarm value	RL to RH PL < PH	Single-precision real number	User
+28 +29	S.PHPL/ S.PHPL2	PL	Lower limit alarm value	RL to RH PL < PH	Single-precision real number	User
+30 +31	S.PHPL/ S.PHPL2	HH	Upper upper limit alarm value	RL to RH PH ≤ HH	Single-precision real number	User
+32 +33	S.PHPL/ S.PHPL2	LL	Lower lower limit alarm value	RL to RH LL ≤ PL	Single-precision real number	User
+38 +39	S.IN	α	Filter coefficient	0 to 1	Single-precision real number	User
+40 +41	S.PHPL/ S.PHPL2	HS	Upper/lower limit alarm hysteresis	0 to 999999 [%]	Single-precision real number	User
+42 +43	S.PHPL/ S.PHPL2	CTIM	Variation rate alarm check time	0 to 999999 [s]	Single-precision real number	User
+44 +45	S.PHPL/ S.PHPL2	DPL	Variation rate alarm value	0 to 100 [%]	Single-precision real number	User
+46 +47	S.R	CT	Control cycle	0 to 999999 [s]	Single-precision real number	User
+48 +49	S.OUT2	DML	Output variation rate limit value	0 to 100 [%]	Single-precision real number	User
+50 +51	S.R	DR	Variation rate limit value	0 to 999999	Single-precision real number	User
+52 +53	S.R	RMAX	Ratio upper limit value	-999999 to 999999	Single-precision real number	User
+54 +55	S.R	RMIN	Ratio lower limit value	-999999 to 999999	Single-precision real number	User
+56 +57	S.R	Rn	Ratio current value	-999999 to 999999	Single-precision real number	System


# 21.10 Processing Time

The following table lists the processing time of process control instructions.

The processing time varies slightly depending on the contents of the source and destination. Assume that the values in the table are reference processing time.

**Point**

When using the file register (R/ZR), module access device (U□\G□), link direct device (J□\□), or module label (only the ones with the label name includes \_D), add extra time described in the section below to each instruction processing time.

 Page 804 Time added to instruction processing time

Instruction name	Condition	Processing time (μs)	
		Minimum	Maximum
S.IN	The loop is running and the ALM bit does not turn on.	32.400	33.800
S.OUT1	The loop is running in AUT mode and the ALM bit does not turn on.	27.500	27.900
S.OUT2	The loop is running in AUT mode and the ALM bit does not turn on.	25.700	26.600
S.MOUT	The loop is running in MAN mode.	19.900	20.400
S.DUTY	Execution cycle = 1 Control output cycle = 10 The loop is running in AUT mode and the ALM bit does not turn on.	29.200	29.900
S.BC	The loop is running in AUT mode and the ALM bit does not turn on.	21.200	21.500
S.PSUM	Integration start signal = On, Integration hold signal = Off	14.700	15.700
S.PID	Set value pattern = 3 (without a cascade) Tracking bit = 0 Execution cycle = Control cycle = 1 Integral constant ≠ 0 Derivative constant ≠ 0 The loop is running in AUT mode and the ALM bit does not turn on.	48.700	49.700
S.2PID	Set value pattern = 3 (without a cascade) Tracking bit = 0 Execution cycle = Control cycle = 1 Integral constant ≠ 0 Derivative constant ≠ 0 The loop is running in AUT mode and the ALM bit does not turn on.	57.300	59.100
S.PIDP	Set value pattern = 3 (without a cascade) Tracking bit = 0 Execution cycle = Control cycle = 1 Integral constant ≠ 0 Derivative constant ≠ 0 The loop is running in AUT mode and the ALM bit does not turn on.	53.200	54.600
S.SPI	Set value pattern = 3 (without a cascade) Tracking bit = 0 Operating time = Sample cycle (ST=STHT) Integral constant ≠ 0 The loop is running in AUT mode and the ALM bit does not turn on.	35.900	36.900
S.IPD	Set value pattern = 3 (without a cascade) Tracking bit = 0 Execution cycle = Control cycle = 1 Integral constant ≠ 0 Derivative constant ≠ 0 The loop is running in AUT mode and the ALM bit does not turn on.	45.600	47.800
S.BPI	Set value pattern = 3 (without a cascade) Tracking bit = 0 Execution cycle = Control cycle = 1 Integral constant ≠ 0 The loop is running in AUT mode and the ALM bit does not turn on.	34.700	35.400
S.R	Set value pattern = 3 (without a cascade) Tracking bit = 0 Execution cycle = Control cycle = 1 The loop is running in AUT mode.	31.300	32.600
S.PHPL	The loop is running in AUT mode and the ALM bit does not turn on.	38.000	39.500

Instruction name	Condition	Processing time (μs)	
		Minimum	Maximum
S.PHPL2	The loop is running in AUT mode and the ALM bit does not turn on.	38.500	41.400
S.LLAG	Input data = 50 with lead-lag compensation Lead time = 1, Lag time = 1	21.100	21.400
S.I	Input data = 50, Integral time = 1 Output initial value = 0	17.100	17.700
S.D	Input data = 50, Derivative time = 1 Output initial value = 0	18.500	18.900
S.DED	Input data = 50 Operation control signal 0 → 1 Data sampling interval = 1 Sampling count = 10 Output initial value = 0 Initial output switching = 0	10.200	10.900
S.HS	Number of inputs = 5 Input data = 50, 100, 150, 200, 250	13.300	13.700
S.LS	Number of inputs = 5 Input data = 50, 100, 150, 200, 250	13.000	13.200
S.MID	Number of inputs = 5 Input data = 50, 100, 150, 200, 250	17.400	17.900
S.AVE	Number of inputs = 2, Input data = 50, 100	16.000	16.300
S.LIMIT	Input data = 50 Upper limit value = 100 Lower limit value = 0 Upper limit hysteresis = 0 Lower limit hysteresis = 0	18.200	18.700
S.VLMT1	Input data = 50 Positive direction limit value = 100 Negative direction limit value = 100 Positive direction hysteresis = 0 Negative direction hysteresis = 0	17.500	17.600
S.VLMT2	Input data = 50 Positive direction limit value = 100 Negative direction limit value = 100 Positive direction hysteresis = 0 Negative direction hysteresis = 0	17.300	17.800
S.ONF2	Input data = 10 Set value pattern = 3 (without a cascade) Tracking bit = 0 Execution cycle = Control cycle = 1 The loop is running in MAN mode.	32.600	34.100
S.ONF3	Input data = 10 Set value pattern = 3 (without a cascade) Tracking bit = 0 Execution cycle = Control cycle = 1 The loop is running in MAN mode.	34.500	35.800
S.DBND	Input data = 50 Dead band upper limit = 100, Dead band lower limit = 0 Input range = 1	17.000	17.600
S.PGS	Number of operation constant break points = 16 Operation type = 0 (Hold type) Execution cycle = 1 Set value = 10 The loop is running in AUT mode and the ALM bit does not turn on.	27.500	28.000
S.SEL	Set value pattern = 18H (E1 and E2 used, without a cascade) Tracking bit = 0 The loop is running in AUT mode and the ALM bit does not turn on.	31.300	32.400
S.BUMP	Output set value = 0, Output control value = 50 Mode switching signal = 1 Delay time = 1, Delay time zone = 1	12.000	12.400
S.AMR	Output addition value = 50, Output subtraction value = 50 Output set value = 0, Operation output signal = 1 Output addition signal = 1, Output subtraction signal = 0 Output upper limit value = 50, Output lower limit value = 0	15.100	15.600

Instruction name	Condition	Processing time (µs)	
		Minimum	Maximum
S.FG	Input data = 50, Number of break points = 2 Break point coordinates (30, 40), (60, 70)	21.600	21.900
S.IFG	Input data = 50, Number of break points = 2 Break point coordinates (30, 40), (60, 70)	20.600	21.300
S.FLT	Input data = 50, Data sampling interval = 1 Sampling count = 10	20.800	21.500
S.SUM	Input data = 50 Input low-cut value = 0, Initial value = 0 Input range = 1	17.500	18.000
S.TPC	When both temperature and pressure are corrected Differential pressure = 100, Measured temperature = 300 Measured pressure = 10000, Design temperature = 0 Bias (temperature) = 273.15 Design pressure = 0 Bias (pressure) = 10332.0	19.300	19.800
S.ENG	Input data = 50, Engineering value upper limit = 100 Engineering value lower limit = 0	18.800	19.000
S.IENG	Input data = 50, Engineering value upper limit = 100 Engineering value lower limit = 0	18.600	18.900
S.ADD	Number of inputs = 2, Input data = 50, 100 Number of coefficients = 2, Coefficient = 1, 1, Bias = 0	17.400	17.800
S.SUB	Number of inputs = 2, Input data = 50, 100 Number of coefficients = 2, Coefficient = 1, 1, Bias = 0	18.800	19.200
S.MUL	Number of inputs = 2, Input data = 50, 100 Number of coefficients = 2, Coefficient = 1, 1, Bias = 0	17.500	17.900
S.DIV	Input data = 50, 100 Coefficient = 1, 1, 1, Bias = 0, 0, 0	18.600	19.000
S.SQR	Input data = 50 Output low-cut value = 0, Coefficient = 10	15.800	16.500
S.ABS	Input data = 50	11.400	11.800
S.>	Input data = 50, 100 Set value = 0, Hysteresis = 0	15.700	16.100
S.<	Input data = 50, 100 Set value = 0, Hysteresis = 0	13.900	14.200
S.=	Input data = 50, 100 Set value = 0	14.000	14.600
S.>=	Input data = 50, 100 Set value = 0, Hysteresis = 0	15.600	16.200
S.<=	Input data = 50, 100 Set value = 0, Hysteresis = 0	13.800	14.100
S.AT1	Set value pattern = 3 (without a cascade) Tracking bit = 0 Execution cycle = 1 The loop is running in MAN mode.	24.600	25.700

## Time added to instruction processing time

When using the file register (R/ZR), module access device (U□\G□), link direct device (J□\□), or module label (only the ones with the label name includes \_D), add extra time described in the section below to each instruction processing time.

Device name		Data type	Device part specification	Additional time (μs)
File register (R/ZR)	When the extended SRAM cassette is not inserted	Bit	Source	0.043
			Destination	0.023
		Word	Source	0.043
			Destination	0.023
		Double word	Source	0.085
			Destination	0.044
	When the extended SRAM cassette is inserted	Bit	Source	0.099
			Destination	0.028
		Word	Source	0.099
			Destination	0.028
		Double word	Source	0.198
			Destination	0.054
Module access device (U□\G□)	Bit	Source	13.000	
		Destination	14.000	
	Word	Source	13.000	
		Destination	14.000	
	Double word	Source	13.000	
		Destination	14.000	
Link direct device (J□\□)	Bit	Source	51.000	
		Destination	52.000	
	Word	Source	51.000	
		Destination	52.000	
	Double word	Source	51.000	
		Destination	52.000	

# 21.11 Number of Basic Steps and Availability of Subset Processing

The number of basic steps and the availability of subset processing are shown below.

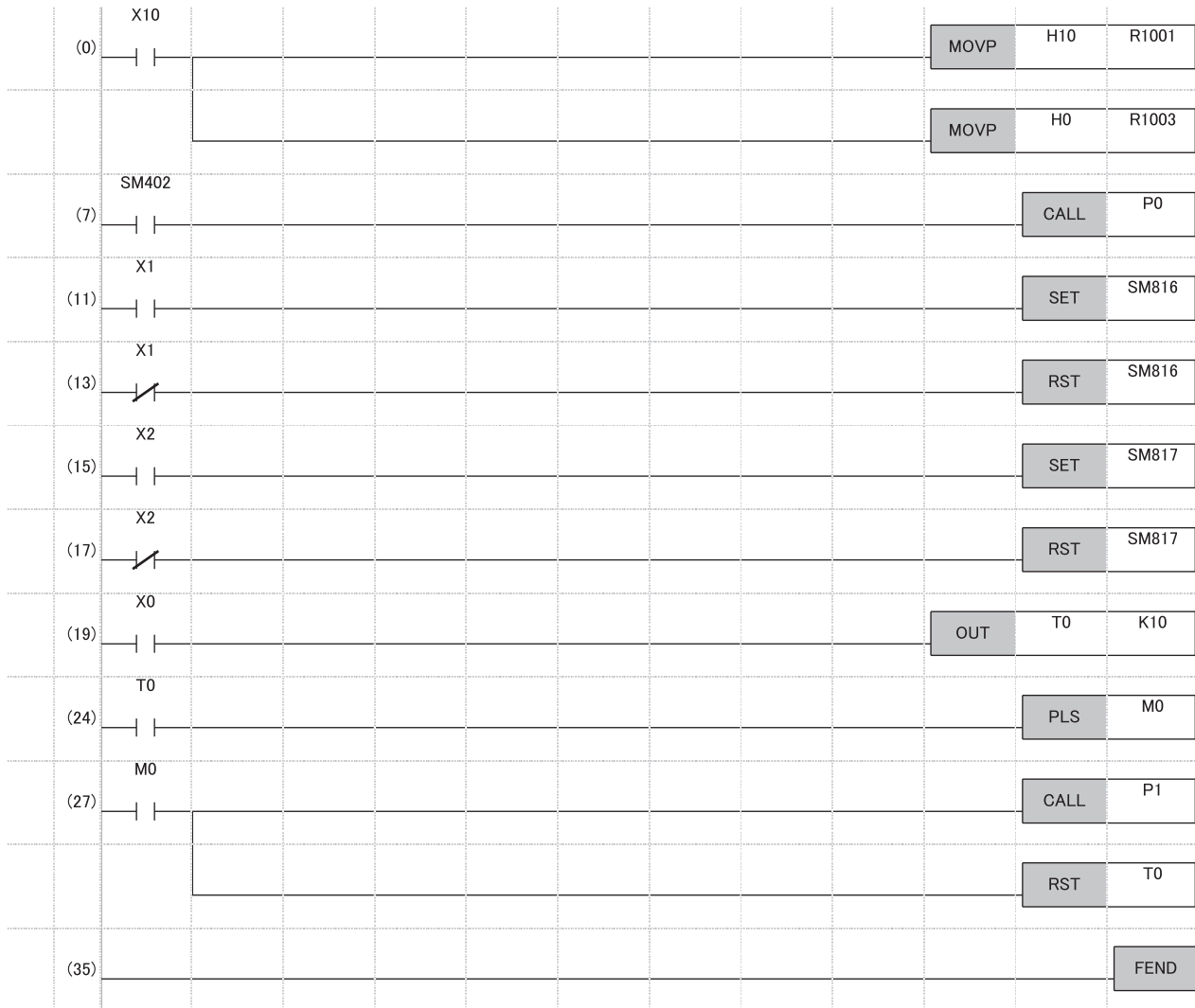
Instruction name	Number of basic steps	Subset availability
S.IN	5	—
S.OUT1	5	—
S.OUT2	5	—
S.MOUT	5	—
S.DUTY	5	—
S.BC	5	—
S.PSUM	5	—
S.PID	6	—
S.2PID	6	—
S.PIDP	6	—
S.SPI	6	—
S.IPD	6	—
S.BPI	6	—
S.R	6	—
S.PHPL	5	—
S.PHPL2	5	—
S.LLAG	5	—
S.I	5	—
S.D	5	—
S.DED	5	—
S.HS	5	—
S.LS	5	—
S.MID	5	—
S.AVE	5	—
S.LIMIT	5	—
S.VLMT1	5	—
S.VLMT2	5	—
S.ONF2	6	—
S.ONF3	6	—
S.DBND	5	—
S.PGS	5	—
S.SEL	6	—
S.BUMP	5	—
S.AMR	5	—
S.FG	5	—
S.IFG	5	—
S.FLT	5	—
S.SUM	5	—
S.TPC	5	—
S.ENG	5	—
S.IENG	5	—
S.ADD	5	—
S.SUB	5	—
S.MUL	5	—
S.DIV	5	—
S.SQR	5	—
S.ABS	5	—

Instruction name	Number of basic steps	Subset availability
S.>	5	—
S.<	5	—
S.=	5	—
S.>=	5	—
S.<=	5	—
S.AT1	6	—



# 21.12 Program Examples

This section provides examples of process control programs using process control instructions. With the following program, the control mode enters the manual mode when the power is turned on. Turning on X10 enters the auto mode (AUTO) and performs PID control.



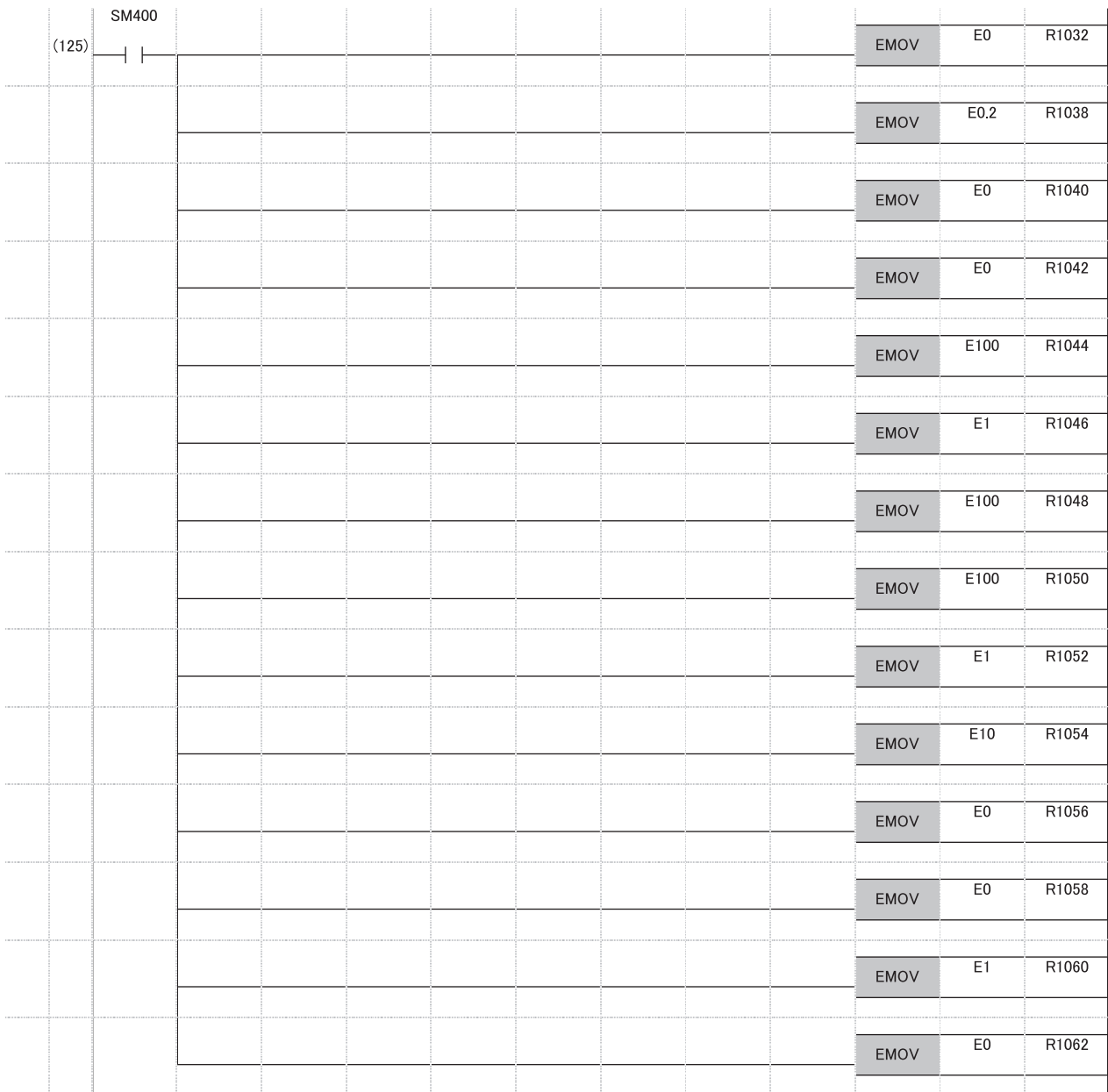
- (0) Set the control mode to AUTO.  
Set the alarm detection (ALM) to 0.
- (7) Jump to the P0 label when SM402 is on.
- (11) Set the last value hold processing.
- (13) Reset the last value hold processing.
- (15) Set the output value hold processing.
- (17) Reset the output value hold processing.
- (19) Adjust to the time set for the execution cycle.
- (27) Jump to the P1 label when M0 is on.  
Turning M0 on clears T0 to 0.
- (35) The end of the sequence program in which SM402 is off is indicated.

P1

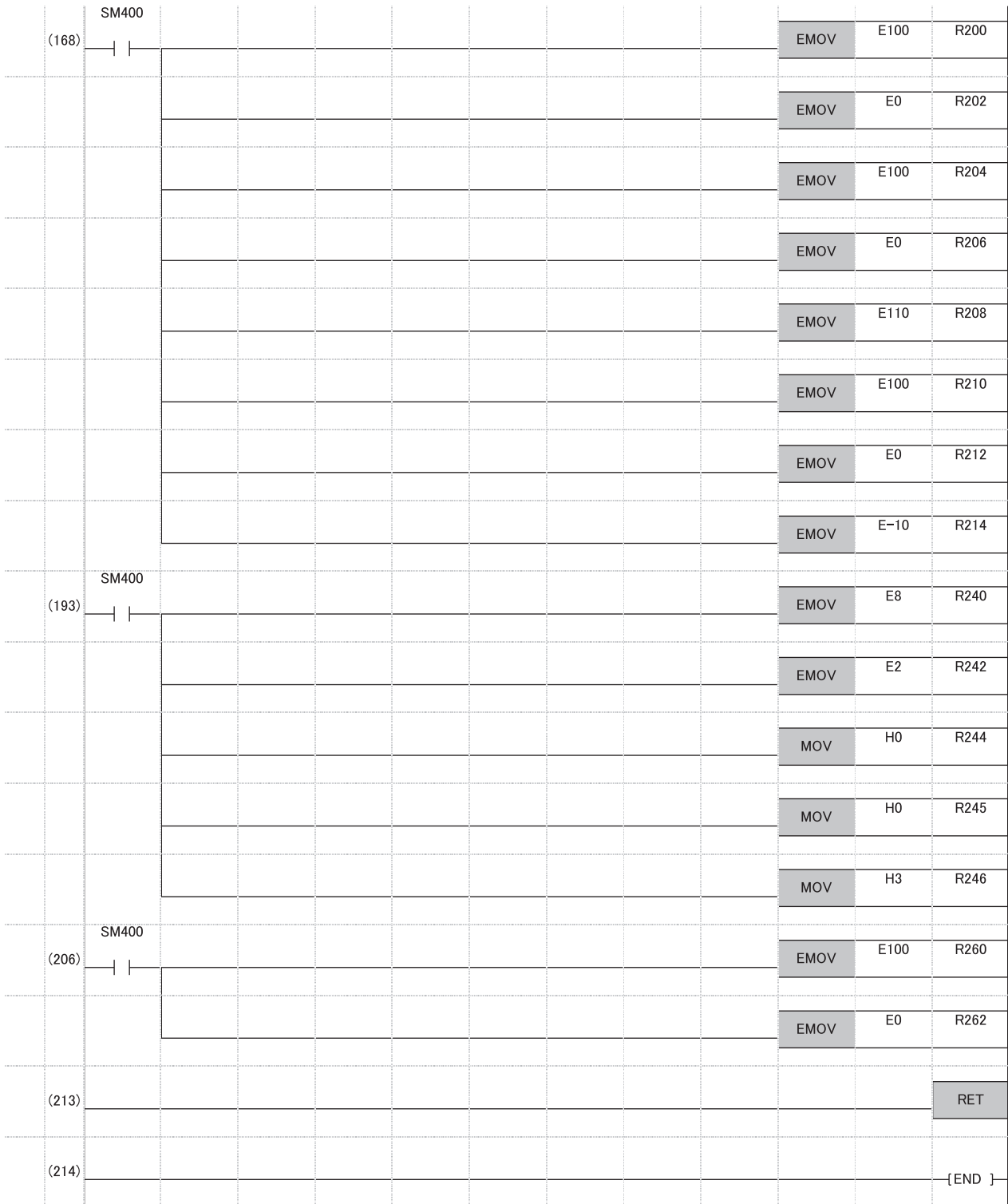
P1	(37)	SM400							INT2FLT	D0	R0
									S.IN	R0	R100 R200 R1000
									EMOV	R100	R20
									S.PHPL	R20	R120 "" R1000
									EMOV	R120	R40
									S.PID	R40 R140	R240 R1000 R300
									EMOV	R140	R60
								S.OUT1	R60	R160 R260 R1000	
								FLT2INT	R160	D1	
	(70)										RET
	(71)										FEND

- (37) Change the D0 value to a single-precision real number and stores it in R0.  
 Set each start device of the S.IN instruction.  
 Transfer the R100 value of the S.IN instruction to R20 of S.PHPL.  
 Set each start device of the S.PHPL instruction.  
 Transfer the R120 value of the S.PHPL instruction to R40 of S.PID.  
 Set the start device of the S.PID instruction.  
 Transfer the R140 value of the S.PID instruction to R60 of S.OUT1.  
 Set each start device of the S.OUT1 instruction.  
 Convert the single-precision real number in R160 and R161 to binary.
- (70) The subroutine program ends.
- (71) The main routine program ends.





(125) Set the default value of the loop tag.






- (168) Set the operation constant of the S.IN instruction.
- (193) Set the operation constant of the S.PID instruction.
- (206) Set the operation constant of the S.OUT1 instruction.
- (213) The subroutine program ends.

# APPENDICES

## Appendix 1 Tag Data List

This section lists the tag data of the process control function blocks.

How to read items in the list is shown below.

Item	Description
Offset	Indicates the offset word of memory data inside the tags.
Label	Indicates tag data (tag member).
Data type	Indicates the memory data structure. <ul style="list-style-type: none"> <li>• BOOL: Bit</li> <li>• WORD: Integer data (word [unsigned]/bit string [16 bits])</li> <li>• INT: Integer data (word [signed])</li> <li>• DINT: Integer data (double word [signed])</li> <li>• REAL: Real number data (single-precision real number)</li> </ul>
Setting/Storage range (Low limit/ High limit)	Indicates the setting range of all items. Refer to following range for PH, PL, HH, LL setting/storage range. <ul style="list-style-type: none"> <li>• PV high limit alarm value (PH<sub>l</sub>): (RL) to (RH) and (PL) &lt; (PH<sub>l</sub>)</li> <li>• PV low limit alarm value (PL): (RL) to (RH) and (PL) &lt; (PH<sub>l</sub>)</li> <li>• PV high high limit alarm value (HH): (RL) to (RH) and (PH<sub>l</sub>) ≤ (HH)</li> <li>• PV low low limit alarm value (LL): (RL) to (RH) and (LL) ≤ (PL)</li> </ul> Set the control cycle (CT) to the integral multiple of an execution cycle. The execution cycle indicates an execution cycle set by GX Works3 and an execution cycle of a program block property.  [Tool] ⇒ [Options] ⇒ [Convert] ⇒ [Process Control Extension Setting]
Initial value	Indicates default value.
Number of digits after the decimal point	Indicates the number of digits after the decimal point. N is indicated by the number of digits after the decimal point of +9[N <sub>l</sub> ].
Unit	Indicates units. UNIT is set in the "Unit Setting" window of PX Developer.  [Monitor Tool Setting] ⇒ [Unit Setting]
Application	Indicates the name/application of tag data (tag member).
Storage	Indicates whether it is allowed to read/write tag data using programs. <ul style="list-style-type: none"> <li>■User It is allowed to read/write tag data. However, tag data with (condition 1/2/3) can only be written only under the following conditions. <ul style="list-style-type: none"> <li>• Condition 1: When changing the control mode using a program, use M+P_MCHG of the tag access FB. Switch to the control mode with TRUE as the corresponding bit item of mode inhibition is not allowed.</li> <li>• Condition 2: Stop alarm (SPA) and output open (OOA) of an alarm are written by a program. When the program sets stop alarm (SPA) as TRUE, stop alarm in the corresponding loop is processed. The bit items except stop alarm (SPA) and output open (OOA) of an alarm are written by the system. Do not write them using a program.</li> <li>• Condition 3: It is allowed to write tag data only when the control mode is MAN or CMV.</li> </ul> </li> <li>■System It is allowed only to read tag data. Do not write tag data. The operations are not guaranteed if it is written. This item is not displayed on "FB Property" window on GX Works3.</li> <li>■Tag data access control Tag data can be written only from the ActiveX container application that uses the tag data access control that is an ActiveX control. For details on tag data access control, refer to the following.   PX Developer Version 1 Operating Manual (Monitor Tool) </li> </ul>
Tag access FB	Indicates the tag access FB which reads/writes tag data.

The following table describes Boolean items of tag data.

Item	Description
Control mode	TRUE: Valid, FALSE: Invalid
Disable Mode Change	TRUE: Valid, FALSE: Invalid
Alarm	TRUE: Detected, FALSE: Reset
Disable alarm detection	TRUE: Valid, FALSE: Invalid
Alarm level	TRUE: Major alarm, FALSE: Minor alarm

# PID

Structure name: M+TM\_PID

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+0	FUNC	INT	1	1	1	—	—	Tag Function Code	System	—
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG M+P_PID(_T) M+P_OUT1 M+P_DUTY
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG M+P_PID(_T) M+P_OUT1 M+P_DUTY
	b5	CAS	BOOL	FALSE	TRUE	FALSE	—	Control Mode: CASCADE	User (condition 1)	M+P_MCHG M+P_PID(_T) M+P_OUT1 M+P_DUTY
	b9	CMV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER MV	User (condition 1)	M+P_MCHG M+P_PID(_T) M+P_OUT1 M+P_DUTY
	bA	CSV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER SV	User (condition 1)	M+P_MCHG M+P_PID(_T) M+P_OUT1 M+P_DUTY
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable MANUAL	User	M+P_MCHG
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable AUTO	User	M+P_MCHG
	b5	CASI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable CASCADE	User	M+P_MCHG
	b9	CMVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER MV	User	M+P_MCHG
	bA	CSVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER SV	User	M+P_MCHG
	bD	ATI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable Auto Tuning	User	M+P_MCHG
	bE	OVRI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable OVERRIDE	User	M+P_MCHG
	bF	SIMI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable SIMULATION	User	M+P_MCHG



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+3	b0	MLA	BOOL	FALSE	TRUE	FALSE	—	—	Output Low Limit Alarm	User (condition 2)	Common
	b1	MHA	BOOL	FALSE	TRUE	FALSE	—	—	Output High Limit Alarm	User (condition 2)	Common
	b2	DVLA	BOOL	FALSE	TRUE	FALSE	—	—	Large Deviation Alarm	User (condition 2)	Common
	b3	DPNA	BOOL	FALSE	TRUE	FALSE	—	—	Negative Variation Rate Alarm	User (condition 2)	Common
	b4	DPPA	BOOL	FALSE	TRUE	FALSE	—	—	Positive Variation Rate Alarm	User (condition 2)	Common
	b5	PLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Limit Alarm	User (condition 2)	Common
	b6	PHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High Limit Alarm	User (condition 2)	Common
	b7	LLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Low Limit Alarm	User (condition 2)	Common
	b8	HHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High High Limit Alarm	User (condition 2)	Common
	b9	SEA	BOOL	FALSE	TRUE	FALSE	—	—	Sensor Error Alarm	User (condition 2)	Common
	bA	OOA	BOOL	FALSE	TRUE	FALSE	—	—	Output Open Alarm	User (condition 2)	Common
	bB	DMLA	BOOL	FALSE	TRUE	FALSE	—	—	Output Variation Rate Limit Alarm	User (condition 2)	Common
	bE	SPA	BOOL	FALSE	TRUE	FALSE	—	—	Stop Alarm	User (condition 2)	Common
+4	b0	MLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output Low Limit Alarm	User	Common
	b1	MHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output High Limit Alarm	User	Common
	b2	DVLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Large Deviation Alarm	User	Common
	b3	DPNI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Negative Variation Rate Alarm	User	Common
	b4	DPPI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Positive Variation Rate Alarm	User	Common
	b5	PLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Limit Alarm	User	Common
	b6	PHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High Limit Alarm	User	Common
	b7	LLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Low Limit Alarm	User	Common
	b8	HHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High High Limit Alarm	User	Common
	b9	SEI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Sensor Error Alarm	User	Common
	bB	DMLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output Variation Rate Limit Alarm	User	Common
	bD	TRKF	BOOL	FALSE	TRUE	FALSE	—	—	Tracking Flag	User	Common
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User	Common



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+5	b0	MLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Low Limit Alarm	User	—
	b1	MHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output High Limit Alarm	User	—
	b2	DVLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Large Deviation Alarm	User	—
	b3	DPNL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Negative Variation Rate Alarm	User	—
	b4	DPPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Positive Variation Rate Alarm	User	—
	b5	PLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Limit Alarm	User	—
	b6	PHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High Limit Alarm	User	—
	b7	LLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Low Limit Alarm	User	—
	b8	HHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High High Limit Alarm	User	—
	b9	SENL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Sensor Error Alarm	User	—
	bA	OOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Open Alarm	User	—
	bB	DMLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Variation Rate Limit Alarm	User	—
bE	SPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Stop Alarm	User	—	
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—	
+8	UNIT	INT	0	127	0	—	—	Unit	User	—	
+9	N_	INT	0	4	1	—	—	No. of digits after the decimal point	User	—	
+10	PV	REAL	RL	RH	0.0	N	UNIT	Process Variable	System	M+P_PHPL	
+11											
+12	MV	REAL	-10	110	0.0	1	%	Manipulated Variable	User (condition 3)	M+P_OUT1 M+P_PID(_T) M+P_DUTY	
+13											
+14	SV	REAL	RL	RH	0.0	N	UNIT	Setting value	User	M+P_PID(_T)	
+15											
+16	DV	REAL	-110	110	0.0	1	%	Deviation	System	M+P_PID(_T)	
+17											
+18	MH	REAL	-10	110	100.0	1	%	MV High Limit Value	User	M+P_OUT1 M+P_PID(_T)	
+19											
+20	ML	REAL	-10	110	0.0	1	%	MV Low Limit Value	User	M+P_OUT1 M+P_PID(_T)	
+21											
+22	RH	REAL	-999999	999999	100.0	N	UNIT	PV Engineering Value High Limit	User	M+P_PHPL	
+23											
+24	RL	REAL	-999999	999999	0.0	N	UNIT	PV Engineering Value Low Limit	User	M+P_PHPL	
+25											
+26	PH_	REAL	RL	RH	100.0	N	UNIT	PV High Limit Alarm Value	User	M+P_PHPL	
+27											

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+28	PL	REAL	RL	RH	0.0	N	UNIT	PV Low Limit Alarm Value	User	M+P_PHPL
+29										
+30	HH	REAL	RL	RH	100.0	N	UNIT	PV High High Limit Alarm Value	User	M+P_PHPL
+31										
+32	LL	REAL	RL	RH	0.0	N	UNIT	PV Low Low Limit Alarm Value	User	M+P_PHPL
+33										
+34	SH	REAL	RL	RH	100.0	N	UNIT	SV High Limit Value	User	—
+35										
+36	SL	REAL	RL	RH	0.0	N	UNIT	SV Low Limit Value	User	—
+37										
+38	ALPHA	REAL	0	1	0.2	2	—	PV Filter Coefficient	User	M+P_IN
+39										
+40	HS	REAL	0	100	0.0	1	%	PV High/Low Limit Alarm Hysteresis	User	M+P_PHPL
+41										
+42	CTIM	REAL	0	9999	0.0	2	s	Variation Rate Alarm Check Time	User	M+P_PHPL
+43										
+44	DPL	REAL	0	100	100.0	1	%	Variation Rate Alarm Value	User	M+P_PHPL
+45										
+46	CT	REAL	0	9999	1.0	2	s	Control Cycle	User	M+P_PID(_T)
+47										
+48	DML	REAL	0	100	100.0	1	%	Output Variation Rate High Limit Value	User	M+P_OUT1 M+P_DUTY
+49										
+50	DVL	REAL	0	100	100.0	1	%	Deviation Limit Value	User	M+P_PID(_T)
+51										
+52	PROPORTIONAL	REAL	0	999	1.0	2	—	Gain	User	M+P_PID(_T)
+53										
+54	INTEGRAL	REAL	0	9999	10.0	1	s	Integral Time	User	M+P_PID(_T)
+55										
+56	DERIVATIVE	REAL	0	9999	0.0	1	s	Derivative Time	User	M+P_PID(_T)
+57										
+58	GW	REAL	0	100	0.0	1	%	Gap Width	User	M+P_PID(_T)
+59										
+60	GG	REAL	0	99	1.0	1	—	Gap Gain	User	M+P_PID(_T)
+61										
+62	MVP	REAL	-999999	999999	0.0	1	—	MV Internal Operation Value	System	—
+63										
+68	CTDUTY	REAL	0	9999	1.0	2	s	Control Output Cycle	User	M+P_DUTY
+69										
+70	AT1STEPMV	REAL	-100	100	0.0	1	%	Step Manipulated Variable for AT1	User	M+P_PID(_T)
+71										
+72	AT1ST	REAL	0	9999	1.0	2	s	Sampling Interval Time for AT1	User	M+P_PID(_T)
+73										
+74	AT1TOUT1	REAL	0	9999	100.0	1	s	Time-out Interval for AT1	User	M+P_PID(_T)
+75										
+76	AT1TOUT2	REAL	0	9999	10.0	1	s	Time-out Interval after Maximum Slope for AT1	User	M+P_PID(_T)
+77										

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+94	b0	DOM_AT_START_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Start Setting	Tag data access control	—
	b1	DOM_AT_STOP_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Stop Setting	Tag data access control	—
	bE	OVR	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: OVERRIDE	Tag data access control	—
	bF	SIM	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: SIMULATION	Tag data access control	—
+95	b0	DIM_AT_RUN	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Tuning	System	—
	b2	DIM_AT_PH	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Input High Limit	System	—
	b3	DIM_AT_PL	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Input Low Limit	System	—
	b4	DIM_AT_MVH	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Output High Limit	System	—
	b5	DIM_AT_MVL	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Output Low Limit	System	—
	b6	DIM_AT_TO	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Time-out	System	—
	b7	DIM_AT_MODE	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Operation Mode	System	—
	b8	DIM_AT_ID	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Identification	System	—

# 2PID

Structure name: M+TM\_2PID

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+0	FUNC	INT	14	14	14	—	—	Tag Function Code	System	—
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG M+P_2PID(_T) M+P_OUT1 M+P_DUTY
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG M+P_2PID(_T) M+P_OUT1 M+P_DUTY
	b5	CAS	BOOL	FALSE	TRUE	FALSE	—	Control Mode: CASCADE	User (condition 1)	M+P_MCHG M+P_2PID(_T) M+P_OUT1 M+P_DUTY
	b9	CMV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER MV	User (condition 1)	M+P_MCHG M+P_2PID(_T) M+P_OUT1 M+P_DUTY
	bA	CSV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER SV	User (condition 1)	M+P_MCHG M+P_2PID(_T) M+P_OUT1 M+P_DUTY
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable MANUAL	User	M+P_MCHG
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable AUTO	User	M+P_MCHG
	b5	CASI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable CASCADE	User	M+P_MCHG
	b9	CMVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER MV	User	M+P_MCHG
	bA	CSVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER SV	User	M+P_MCHG
	bD	ATI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable Auto Tuning	User	M+P_MCHG
	bE	OVRI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable OVERRIDE	User	M+P_MCHG
	bF	SIMI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable SIMULATION	User	M+P_MCHG

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+3	b0	MLA	BOOL	FALSE	TRUE	FALSE	—	—	Output Low Limit Alarm	User (condition 2)	Common
	b1	MHA	BOOL	FALSE	TRUE	FALSE	—	—	Output High Limit Alarm	User (condition 2)	Common
	b2	DVLA	BOOL	FALSE	TRUE	FALSE	—	—	Large Deviation Alarm	User (condition 2)	Common
	b3	DPNA	BOOL	FALSE	TRUE	FALSE	—	—	Negative Variation Rate Alarm	User (condition 2)	Common
	b4	DPPA	BOOL	FALSE	TRUE	FALSE	—	—	Positive Variation Rate Alarm	User (condition 2)	Common
	b5	PLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Limit Alarm	User (condition 2)	Common
	b6	PHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High Limit Alarm	User (condition 2)	Common
	b7	LLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Low Limit Alarm	User (condition 2)	Common
	b8	HHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High High Limit Alarm	User (condition 2)	Common
	b9	SEA	BOOL	FALSE	TRUE	FALSE	—	—	Sensor Error Alarm	User (condition 2)	Common
	bA	OOA	BOOL	FALSE	TRUE	FALSE	—	—	Output Open Alarm	User (condition 2)	Common
	bB	DMLA	BOOL	FALSE	TRUE	FALSE	—	—	Output Variation Rate Limit Alarm	User (condition 2)	Common
	bE	SPA	BOOL	FALSE	TRUE	FALSE	—	—	Stop Alarm	User (condition 2)	Common
+4	b0	MLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output Low Limit Alarm	User	Common
	b1	MHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output High Limit Alarm	User	Common
	b2	DVLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Large Deviation Alarm	User	Common
	b3	DPNI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Negative Variation Rate Alarm	User	Common
	b4	DPPI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Positive Variation Rate Alarm	User	Common
	b5	PLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Limit Alarm	User	Common
	b6	PHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High Limit Alarm	User	Common
	b7	LLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Low Limit Alarm	User	Common
	b8	HHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High High Limit Alarm	User	Common
	b9	SEI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Sensor Error Alarm	User	Common
	bB	DMLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output Variation Rate Limit Alarm	User	Common
	bD	TRKF	BOOL	FALSE	TRUE	FALSE	—	—	Tracking Flag	User	Common
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User	Common



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+5	b0	MLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Low Limit Alarm	User	—
	b1	MHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output High Limit Alarm	User	—
	b2	DVLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Large Deviation Alarm	User	—
	b3	DPNL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Negative Variation Rate Alarm	User	—
	b4	DPPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Positive Variation Rate Alarm	User	—
	b5	PLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Limit Alarm	User	—
	b6	PHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High Limit Alarm	User	—
	b7	LLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Low Limit Alarm	User	—
	b8	HHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High High Limit Alarm	User	—
	b9	SENL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Sensor Error Alarm	User	—
	bA	OOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Open Alarm	User	—
	bB	DMLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Variation Rate Limit Alarm	User	—
bE	SPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Stop Alarm	User	—	
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—	
+8	UNIT	INT	0	127	0	—	—	Unit	User	—	
+9	N_	INT	0	4	1	—	—	No. of digits after the decimal point	User	—	
+10	PV	REAL	RL	RH	0.0	N	UNIT	Process Variable	System	M+P_PHPL	
+11											
+12	MV	REAL	-10	110	0.0	1	%	Manipulated Variable	User (condition 3)	M+P_OUT1 M+P_2PID(_T) M+P_DUTY	
+13											
+14	SV	REAL	RL	RH	0.0	N	UNIT	Setting value	User	M+P_2PID(_T)	
+15											
+16	DV	REAL	-110	110	0.0	1	%	Deviation	System	M+P_2PID(_T)	
+17											
+18	MH	REAL	-10	110	100.0	1	%	MV High Limit Value	User	M+P_OUT1 M+P_2PID(_T)	
+19											
+20	ML	REAL	-10	110	0.0	1	%	MV Low Limit Value	User	M+P_OUT1 M+P_2PID(_T)	
+21											
+22	RH	REAL	-999999	999999	100.0	N	UNIT	PV Engineering Value High Limit	User	M+P_PHPL	
+23											
+24	RL	REAL	-999999	999999	0.0	N	UNIT	PV Engineering Value Low Limit	User	M+P_PHPL	
+25											
+26	PH_	REAL	RL	RH	100.0	N	UNIT	PV High Limit Alarm Value	User	M+P_PHPL	
+27											

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+28	PL	REAL	RL	RH	0.0	N	UNIT	PV Low Limit Alarm Value	User	M+P_PHPL
+29										
+30	HH	REAL	RL	RH	100.0	N	UNIT	PV High High Limit Alarm Value	User	M+P_PHPL
+31										
+32	LL	REAL	RL	RH	0.0	N	UNIT	PV Low Low Limit Alarm Value	User	M+P_PHPL
+33										
+34	SH	REAL	RL	RH	100.0	N	UNIT	SV High Limit Value	User	—
+35										
+36	SL	REAL	RL	RH	0.0	N	UNIT	SV Low Limit Value	User	—
+37										
+38	ALPHA	REAL	0	1	0.2	2	—	PV Filter Coefficient	User	M+P_IN
+39										
+40	HS	REAL	0	100	0.0	1	%	PV High/Low Limit Alarm Hysteresis	User	M+P_PHPL
+41										
+42	CTIM	REAL	0	9999	0.0	2	s	Variation Rate Alarm Check Time	User	M+P_PHPL
+43										
+44	DPL	REAL	0	100	100.0	1	%	Variation Rate Alarm Value	User	M+P_PHPL
+45										
+46	CT	REAL	0	9999	1.0	2	s	Control Cycle	User	M+P_2PID(_T)
+47										
+48	DML	REAL	0	100	100.0	1	%	Output Variation Rate High Limit Value	User	M+P_OUT1
+49										
+50	DVL	REAL	0	100	100.0	1	%	Deviation Limit Value	User	M+P_2PID(_T)
+51										
+52	PROPORTIONAL	REAL	0	999	1.0	2	—	Gain	User	M+P_2PID(_T)
+53										
+54	INTEGRAL	REAL	0	9999	10.0	1	s	Integral Time	User	M+P_2PID(_T)
+55										
+56	DERIVATIVE	REAL	0	9999	0.0	1	s	Derivative Time	User	M+P_2PID(_T)
+57										
+58	GW	REAL	0	100	0.0	1	%	Gap Width	User	M+P_2PID(_T)
+59										
+60	GG	REAL	0	99	1.0	1	—	Gap Gain	User	M+P_2PID(_T)
+61										
+62	MVP	REAL	-999999	999999	0.0	1	—	MV Internal Operation Value	System	—
+63										
+64	ALPHA2	REAL	0	1	0.0	2	—	2-degree-of-freedom Parameter Alpha	User	M+P_2PID(_T)
+65										
+66	BETA2	REAL	0	1	1.0	2	—	2-degree-of-freedom Parameter Beta	User	M+P_2PID(_T)
+67										
+68	CTDUTY	REAL	0	9999	1.0	2	s	Control Output Cycle	User	M+P_DUTY
+69										
+70	AT1STEPMV	REAL	-100	100	0.0	1	%	Step Manipulated Variable for AT1	User	M+P_2PID(_T)
+71										
+72	AT1ST	REAL	0	9999	1.0	2	s	Sampling Interval Time for AT1	User	M+P_2PID(_T)
+73										
+74	AT1TOUT1	REAL	0	9999	100.0	1	s	Time-out Interval for AT1	User	M+P_2PID(_T)
+75										



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+76	AT1TOUT2	REAL	0	9999	10.0	1	s	Time-out Interval after Maximum Slope for AT1	User	M+P_2PID(_T)	
+77											
+94	b0	DOM_AT_START_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Start Setting	Tag data access control	—
	b1	DOM_AT_STOP_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Stop Setting	Tag data access control	—
	bE	OVR	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: OVERRIDE	Tag data access control	—
	bF	SIM	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: SIMULATION	Tag data access control	—
+95	b0	DIM_AT_RUN	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Tuning	System	—
	b2	DIM_AT_PH	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Input High Limit	System	—
	b3	DIM_AT_PL	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Input Low Limit	System	—
	b4	DIM_AT_MVH	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Output High Limit	System	—
	b5	DIM_AT_MVL	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Output Low Limit	System	—
	b6	DIM_AT_TO	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Time-out	System	—
	b7	DIM_AT_MODE	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Operation Mode	System	—
	b8	DIM_AT_ID	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Identification	System	—



# 2PIDH

Structure name: M+TM\_2PIDH

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+0	FUNC	INT	17	17	17	—	—	Tag Function Code	System	—	
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG M+P_2PIDH(_T)_ M+P_OUT3
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG M+P_2PIDH(_T)_ M+P_OUT3
	b5	CAS	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: CASCADE	User (condition 1)	M+P_MCHG M+P_2PIDH(_T)_ M+P_OUT3
	b9	CMV	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: COMPUTER MV	User (condition 1)	M+P_MCHG M+P_2PIDH(_T)_ M+P_OUT3
	bA	CSV	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: COMPUTER SV	User (condition 1)	M+P_MCHG M+P_2PIDH(_T)_ M+P_OUT3
	bB	CASDR	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: CASCADE DIRECT	User (condition 1)	M+P_MCHG M+P_2PIDH(_T)_ M+P_OUT3
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable MANUAL	User	M+P_MCHG
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable AUTO	User	M+P_MCHG
	b5	CASI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable CASCADE	User	M+P_MCHG
	b9	CMVI	BOOL	FALSE	TRUE	TRUE	—	—	Disable Mode Change: Disable COMPUTER MV	User	M+P_MCHG
	bA	CSVI	BOOL	FALSE	TRUE	TRUE	—	—	Disable Mode Change: Disable COMPUTER SV	User	M+P_MCHG
	bB	CASDRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable CASCADE DIRECT	User	M+P_MCHG
	bC	TSTPI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable TAG STOP	User	M+P_MCHG
	bD	ATI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable Auto Tuning	User	M+P_MCHG
	bE	OVRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable OVERRIDE	User	M+P_MCHG
	bF	SIMI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable SIMULATION	User	M+P_MCHG



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+3	b0	MLA	BOOL	FALSE	TRUE	FALSE	—	—	Output Low Limit Alarm	User (condition 2)	Common
	b1	MHA	BOOL	FALSE	TRUE	FALSE	—	—	Output High Limit Alarm	User (condition 2)	Common
	b2	DVLA	BOOL	FALSE	TRUE	FALSE	—	—	Large Deviation Alarm	User (condition 2)	Common
	b3	DPNA	BOOL	FALSE	TRUE	FALSE	—	—	Negative Variation Rate Alarm	User (condition 2)	Common
	b4	DPPA	BOOL	FALSE	TRUE	FALSE	—	—	Positive Variation Rate Alarm	User (condition 2)	Common
	b5	PLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Limit Alarm	User (condition 2)	Common
	b6	PHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High Limit Alarm	User (condition 2)	Common
	b7	LLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Low Limit Alarm	User (condition 2)	Common
	b8	HHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High High Limit Alarm	User (condition 2)	Common
	b9	SEA	BOOL	FALSE	TRUE	FALSE	—	—	Sensor Error Alarm	User (condition 2)	Common
	bA	OOA	BOOL	FALSE	TRUE	FALSE	—	—	Output Open Alarm	User (condition 2)	Common
	bB	DMLA	BOOL	FALSE	TRUE	FALSE	—	—	Output Variation Rate Limit Alarm	User (condition 2)	Common
	bE	SPA	BOOL	FALSE	TRUE	FALSE	—	—	Stop Alarm	User (condition 2)	Common
+4	b0	MLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output Low Limit Alarm	User	Common
	b1	MHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output High Limit Alarm	User	Common
	b2	DVLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Large Deviation Alarm	User	Common
	b3	DPNI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Negative Variation Rate Alarm	User	Common
	b4	DPPI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Positive Variation Rate Alarm	User	Common
	b5	PLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Limit Alarm	User	Common
	b6	PHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High Limit Alarm	User	Common
	b7	LLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Low Limit Alarm	User	Common
	b8	HHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High High Limit Alarm	User	Common
	b9	SEI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Sensor Error Alarm	User	Common
	bB	DMLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output Variation Rate Limit Alarm	User	Common
	bD	TRKF	BOOL	FALSE	TRUE	FALSE	—	—	Tracking Flag	User	Common
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User	Common

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+5	b0	MLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Low Limit Alarm	User	—
	b1	MHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output High Limit Alarm	User	—
	b2	DVLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Large Deviation Alarm	User	—
	b3	DPNL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Negative Variation Rate Alarm	User	—
	b4	DPPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Positive Variation Rate Alarm	User	—
	b5	PLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Limit Alarm	User	—
	b6	PHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High Limit Alarm	User	—
	b7	LLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Low Limit Alarm	User	—
	b8	HHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High High Limit Alarm	User	—
	b9	SENL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Sensor Error Alarm	User	—
	bA	OOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Open Alarm	User	—
	bB	DMLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Variation Rate Limit Alarm	User	—
	bE	SPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Stop Alarm	User	—
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—	
+8	UNIT	INT	0	127	0	—	—	Unit	User	—	
+9	N_	INT	0	4	1	—	—	No. of digits after the decimal point	User	—	
+10	PV	REAL	RL	RH	0.0	N	UNIT	Process Variable	System	M+P_PHPL	
+11											
+12	MV	REAL	-10	110	0.0	1	%	Manipulated Variable	User (condition 3)	M+P_OUT3 M+P_2PIDH(_T)_	
+13											
+14	SVC	REAL	RL	RH	0.0	N	UNIT	Setting Value (Current)	System	M+P_2PIDH(_T)_	
+15											
+16	DV	REAL	-110	110	0.0	1	%	Deviation	System	M+P_2PIDH(_T)_	
+17											
+18	MH	REAL	-10	110	100.0	1	%	MV High Limit Value	User	M+P_OUT3 M+P_2PIDH(_T)_	
+19											
+20	ML	REAL	-10	110	0.0	1	%	MV Low Limit Value	User	M+P_OUT3 M+P_2PIDH(_T)_	
+21											
+22	RH	REAL	-999999	999999	100.0	N	UNIT	PV Engineering Value High Limit	User	M+P_PHPL	
+23											

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+24	RL	REAL			0.0	N	UNIT	PV Engineering Value Low Limit	User	M+P_PHPL
+25										
+26	PH_	REAL	RL	RH	100.0	N	UNIT	PV High Limit Alarm Value	User	M+P_PHPL
+27										
+28	PL	REAL	RL	RH	0.0	N	UNIT	PV Low Limit Alarm Value	User	M+P_PHPL
+29										
+30	HH	REAL	RL	RH	100.0	N	UNIT	PV High High Limit Alarm Value	User	M+P_PHPL
+31										
+32	LL	REAL	RL	RH	0.0	N	UNIT	PV Low Low Limit Alarm Value	User	M+P_PHPL
+33										
+34	SH	REAL	RL	RH	100.0	N	UNIT	SV High Limit Value	User	M+P_2PIDH(_T)_
+35										
+36	SL	REAL	RL	RH	0.0	N	UNIT	SV Low Limit Value	User	M+P_2PIDH(_T)_
+37										
+38	ALPHA	REAL	0	1	0.2	2	—	PV Filter Coefficient	User	M+P_IN
+39										
+40	HS	REAL	0	100	0.0	1	%	PV High/Low Limit Alarm Hysteresis	User	M+P_PHPL
+41										
+42	CTIM	REAL	0	9999	0.0	2	s	Variation Rate Alarm Check Time	User	M+P_PHPL
+43										
+44	DPL	REAL	0	100	100.0	1	%	Variation Rate Alarm Value	User	M+P_PHPL
+45										
+46	CT	REAL	0	9999	1.0	2	s	Control Cycle	User	M+P_2PIDH(_T)_
+47										
+48	DML	REAL	0	100	100.0	1	%	Output Variation Rate High Limit Value	User	M+P_OUT3
+49										
+50	DVL	REAL	0	100	100.0	1	%	Deviation Limit Value	User	M+P_2PIDH(_T)_
+51										
+52	PROPORTIONAL	REAL	0	999	1.0	2	—	Gain	User	M+P_2PIDH(_T)_
+53										
+54	INTEGRAL	REAL	0	9999	10.0	1	s	Integral Time	User	M+P_2PIDH(_T)_
+55										
+56	DERIVATIVE	REAL	0	9999	0.0	1	s	Derivative Time	User	M+P_2PIDH(_T)_
+57										
+58	GW	REAL	0	100	0.0	1	—	Gap Width	User	M+P_2PIDH(_T)_
+59										
+60	GG	REAL	0	99	1.0	1	—	Gap Gain	User	M+P_2PIDH(_T)_
+61										
+62	MVP	REAL	-999999	999999	0.0	1	—	MV Internal Operation Value	System	—
+63										
+64	ALPHA2	REAL	0	1	0.0	2	—	2-degree-of-freedom Parameter Alpha	User	M+P_2PIDH(_T)_
+65										
+66	BETA2	REAL	0	1	1.0	2	—	2-degree-of-freedom Parameter Beta	User	M+P_2PIDH(_T)_
+67										
+68	CTDUTY	REAL	0	9999	1.0	2	s	Control Output Cycle	User	Reserved for future use
+69										
+70	AT1STEPMV	REAL	-100	100	0.0	1	%	Step Manipulated Variable for AT1	User	M+P_2PIDH(_T)_
+71										

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+72	AT1ST	REAL	0	9999	1.0	2	s	Sampling Interval Time for AT1	User	M+P_2PIDH(_T)_	
+73											
+74	AT1TOUT1	REAL	0	9999	100.0	1	s	Time-out Interval for AT1/AT2	User	M+P_2PIDH(_T)_	
+75											
+76	AT1TOUT2	REAL	0	9999	10.0	1	s	Time-out Interval after Maximum Slope for AT1	User	M+P_2PIDH(_T)_	
+77											
+78	AT2HS	REAL	0	10	1.0	1	%	Hysteresis for AT2	User	M+P_2PIDH(_T)_	
+79											
+80	AT2MVH	REAL	0	100	100.0	1	%	Output High Limit Value for AT2	User	M+P_2PIDH(_T)_	
+81											
+82	AT2MVL	REAL	0	100	0.0	1	%	Output Low Limit Value for AT2	User	M+P_2PIDH(_T)_	
+83											
+86	ATTTYPE	INT	0	4	1	—	—	Control Type for AT	User	M+P_2PIDH(_T)_	
+87	bD	SVLA	BOOL	FALSE	TRUE	FALSE	—	—	SV Low Alarm	System	Common
	bE	SVHA	BOOL	FALSE	TRUE	FALSE	—	—	SV High Alarm	System	Common
	bF	DSVLA	BOOL	FALSE	TRUE	FALSE	—	—	SV Variation Rate Limit Alarm	System	Common
+88	bD	SVLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable SV Low Alarm	User	Common
	bE	SVHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable SV High Alarm	User	Common
	bF	DSVLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable SV Variation Rate Limit Alarm	User	Common
+89	bD	SVLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of SV Low Alarm	User	Common
	bE	SVHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of SV High Alarm	User	Common
	bF	DSVLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of SV Variation Rate Limit Alarm	User	Common
+90	SV	REAL	RL	RH	0.0	N	UNIT	Setting Value (Target)	User	—	
+91											
+92	DSVL	REAL	0	100	100.0	1	%	SV Variation Rate High Limit Value	User	M+P_2PIDH(_T)_	
+93											
+94	b0	DOM_AT_START_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Start Setting	Tag data access control	—
	b1	DOM_AT_STOP_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Stop Setting	Tag data access control	—
	b2	DOM_AT_TYPE	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Command Type	Tag data access control	—
	bD	TSTP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: TAG STOP	Tag data access control	—
	bE	OVR	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: OVERRIDE	Tag data access control	—
	bF	SIM	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: SIMULATION	Tag data access control	—



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+95	b0	DIM_AT_RUN	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Tuning	System	—
	b2	DIM_AT_PH	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Input High Limit	System	—
	b3	DIM_AT_PL	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Input Low Limit	System	—
	b4	DIM_AT_MVH	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Output High Limit	System	—
	b5	DIM_AT_MVL	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Output Low Limit	System	—
	b6	DIM_AT_TO	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Time-out	System	—
	b7	DIM_AT_MODE	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Operation Mode	System	—
	b8	DIM_AT_ID	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Identification	System	—
	b9	DIM_PREMV	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Preset MV	System	—
	bA	DIM_MVHLD	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: MV Hold	System	—
	bB	DIM_MVTRK	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: MV Tracking	System	—

# PIDP

Structure name: M+TM\_PIDP

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+0	FUNC	INT	2	2	2	—	—	Tag Function Code	System	—	
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG M+P_PIDP(_T) <sup>*1</sup>
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG M+P_PIDP(_T) <sup>*1</sup>
	b5	CAS	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: CASCADE	User (condition 1)	M+P_MCHG M+P_PIDP(_T) <sup>*1</sup>
	b9	CMV	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: COMPUTER MV	User (condition 1)	M+P_MCHG M+P_PIDP(_T) <sup>*1</sup>
	bA	CSV	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: COMPUTER SV	User (condition 1)	M+P_MCHG M+P_PIDP(_T) <sup>*1</sup>
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable MANUAL	User	M+P_MCHG
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable AUTO	User	M+P_MCHG
	b5	CASI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable CASCADE	User	M+P_MCHG
	b9	CMVI	BOOL	FALSE	TRUE	TRUE	—	—	Disable Mode Change: Disable COMPUTER MV	User	M+P_MCHG
	bA	CSVI	BOOL	FALSE	TRUE	TRUE	—	—	Disable Mode Change: Disable COMPUTER SV	User	M+P_MCHG
	bE	OVRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable OVERRIDE	User	M+P_MCHG
	bF	SIMI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable SIMULATION	User	M+P_MCHG



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+3	b0	MLA	BOOL	FALSE	TRUE	FALSE	—	—	Output Low Limit Alarm	User (condition 2)	Common
	b1	MHA	BOOL	FALSE	TRUE	FALSE	—	—	Output High Limit Alarm	User (condition 2)	Common
	b2	DVLA	BOOL	FALSE	TRUE	FALSE	—	—	Large Deviation Alarm	User (condition 2)	Common
	b3	DPNA	BOOL	FALSE	TRUE	FALSE	—	—	Negative Variation Rate Alarm	User (condition 2)	Common
	b4	DPPA	BOOL	FALSE	TRUE	FALSE	—	—	Positive Variation Rate Alarm	User (condition 2)	Common
	b5	PLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Limit Alarm	User (condition 2)	Common
	b6	PHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High Limit Alarm	User (condition 2)	Common
	b7	LLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Low Limit Alarm	User (condition 2)	Common
	b8	HHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High High Limit Alarm	User (condition 2)	Common
	b9	SEA	BOOL	FALSE	TRUE	FALSE	—	—	Sensor Error Alarm	User (condition 2)	Common
	bA	OOA	BOOL	FALSE	TRUE	FALSE	—	—	Output Open Alarm	User (condition 2)	Common
	bB	DMLA	BOOL	FALSE	TRUE	FALSE	—	—	Output Variation Rate Limit Alarm	User (condition 2)	Common
	bE	SPA	BOOL	FALSE	TRUE	FALSE	—	—	Stop Alarm	User (condition 2)	Common
+4	b0	MLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output Low Limit Alarm	User	Common
	b1	MHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output High Limit Alarm	User	Common
	b2	DVLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Large Deviation Alarm	User	Common
	b3	DPNI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Negative Variation Rate Alarm	User	Common
	b4	DPPI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Positive Variation Rate Alarm	User	Common
	b5	PLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Limit Alarm	User	Common
	b6	PHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High Limit Alarm	User	Common
	b7	LLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Low Limit Alarm	User	Common
	b8	HHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High High Limit Alarm	User	Common
	b9	SEI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Sensor Error Alarm	User	Common
	bB	DMLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output Variation Rate Limit Alarm	User	Common
	bD	TRKF	BOOL	FALSE	TRUE	FALSE	—	—	Tracking Flag	User	Common
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User	Common



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+5	b0	MLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Low Limit Alarm	User	—
	b1	MHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output High Limit Alarm	User	—
	b2	DVLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Large Deviation Alarm	User	—
	b3	DPNL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Negative Variation Rate Alarm	User	—
	b4	DPPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Positive Variation Rate Alarm	User	—
	b5	PLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Limit Alarm	User	—
	b6	PHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High Limit Alarm	User	—
	b7	LLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Low Limit Alarm	User	—
	b8	HHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High High Limit Alarm	User	—
	b9	SENL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Sensor Error Alarm	User	—
	bA	OOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Open Alarm	User	—
	bB	DMLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Variation Rate Limit Alarm	User	—
	bE	SPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Stop Alarm	User	—
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—	
+8	UNIT	INT	0	127	0	—	—	Unit	User	—	
+9	N_	INT	0	4	1	—	—	No. of digits after the decimal point	User	—	
+10	PV	REAL	RL	RH	0.0	N	UNIT	Process Variable	System	M+P_PHPL	
+11											
+12	MV	REAL	-10	110	0.0	1	%	Manipulated Variable	User (condition 3)	M+P_PIDP(_T) <sup>1</sup>	
+13											
+14	SV	REAL	RL	RH	0.0	N	UNIT	Setting value	User	M+P_PIDP(_T) <sup>1</sup>	
+15											
+16	DV	REAL	-110	110	0.0	1	%	Deviation	System	M+P_PIDP(_T) <sup>1</sup>	
+17											
+18	MH	REAL	-10	110	100.0	1	%	MV High Limit Value	User	M+P_PIDP(_T) <sup>1</sup>	
+19											
+20	ML	REAL	-10	110	0.0	1	%	MV Low Limit Value	User	M+P_PIDP(_T) <sup>1</sup>	
+21											
+22	RH	REAL	-999999	999999	100.0	N	UNIT	PV Engineering Value High Limit	User	M+P_PHPL	
+23											

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+24	RL	REAL	-999999	999999	0.0	N	UNIT	PV Engineering Value Low Limit	User	M+P_PHPL	
+25											
+26	PH_	REAL	RL	RH	100.0	N	UNIT	PV High Limit Alarm Value	User	M+P_PHPL	
+27											
+28	PL	REAL	RL	RH	0.0	N	UNIT	PV Low Limit Alarm Value	User	M+P_PHPL	
+29											
+30	HH	REAL	RL	RH	100.0	N	UNIT	PV High High Limit Alarm Value	User	M+P_PHPL	
+31											
+32	LL	REAL	RL	RH	0.0	N	UNIT	PV Low Low Limit Alarm Value	User	M+P_PHPL	
+33											
+34	SH	REAL	RL	RH	100.0	N	UNIT	SV High Limit Value	User	—	
+35											
+36	SL	REAL	RL	RH	0.0	N	UNIT	SV Low Limit Value	User	—	
+37											
+38	ALPHA	REAL	0	1	0.2	2	—	PV Filter Coefficient	User	M+P_IN	
+39											
+40	HS	REAL	0	100	0.0	1	%	PV High/Low Limit Alarm Hysteresis	User	M+P_PHPL	
+41											
+42	CTIM	REAL	0	9999	0.0	2	s	Variation Rate Alarm Check Time	User	M+P_PHPL	
+43											
+44	DPL	REAL	0	100	100.0	1	%	Variation Rate Alarm Value	User	M+P_PHPL	
+45											
+46	CT	REAL	0	9999	1.0	2	s	Control Cycle	User	M+P_PIDP(_T)* <sup>1</sup>	
+47											
+48	DML	REAL	0	100	100.0	1	%	Output Variation Rate High Limit Value	User	M+P_PIDP(_T)* <sup>1</sup>	
+49											
+50	DVL	REAL	0	100	100.0	1	%	Deviation Limit Value	User	M+P_PIDP(_T)* <sup>1</sup>	
+51											
+52	PROPORTIONAL	REAL	0	999	1.0	2	—	Gain	User	M+P_PIDP(_T)* <sup>1</sup>	
+53											
+54	INTEGRAL	REAL	0	9999	10.0	1	s	Integral Time	User	M+P_PIDP(_T)* <sup>1</sup>	
+55											
+56	DERIVATIVE	REAL	0	9999	0.0	1	s	Derivative Time	User	M+P_PIDP(_T)* <sup>1</sup>	
+57											
+58	GW	REAL	0	100	0.0	1	%	Gap Width	User	M+P_PIDP(_T)* <sup>1</sup>	
+59											
+60	GG	REAL	0	99	1.0	1	—	Gap Gain	User	M+P_PIDP(_T)* <sup>1</sup>	
+61											
+94	bE	OVR	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: OVERRIDE	System	—
	bF	SIM	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: SIMULATION	System	—

\*1 "M+P\_PIDP\_EX(\_T)\_" is included.

# SPI

Structure name: M+TM\_SPI

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+0	FUNC	INT	3	3	3	—	—	Tag Function Code	System	—
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG M+P_SPI(_T) M+P_OUT1 M+P_DUTY
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG M+P_SPI(_T) M+P_OUT1 M+P_DUTY
	b5	CAS	BOOL	FALSE	TRUE	FALSE	—	Control Mode: CASCADE	User (condition 1)	M+P_MCHG M+P_SPI(_T) M+P_OUT1 M+P_DUTY
	b9	CMV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER MV	User (condition 1)	M+P_MCHG M+P_SPI(_T) M+P_OUT1 M+P_DUTY
	bA	CSV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER SV	User (condition 1)	M+P_MCHG M+P_SPI(_T) M+P_OUT1 M+P_DUTY
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable MANUAL	User	M+P_MCHG
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable AUTO	User	M+P_MCHG
	b5	CASI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable CASCADE	User	M+P_MCHG
	b9	CMVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER MV	User	M+P_MCHG
	bA	CSVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER SV	User	M+P_MCHG
	bE	OVRI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable OVERRIDE	User	M+P_MCHG
	bF	SIMI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable SIMULATION	User	M+P_MCHG



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+3	b0	MLA	BOOL	FALSE	TRUE	FALSE	—	—	Output Low Limit Alarm	User (condition 2)	Common
	b1	MHA	BOOL	FALSE	TRUE	FALSE	—	—	Output High Limit Alarm	User (condition 2)	Common
	b2	DVLA	BOOL	FALSE	TRUE	FALSE	—	—	Large Deviation Alarm	User (condition 2)	Common
	b3	DPNA	BOOL	FALSE	TRUE	FALSE	—	—	Negative Variation Rate Alarm	User (condition 2)	Common
	b4	DPPA	BOOL	FALSE	TRUE	FALSE	—	—	Positive Variation Rate Alarm	User (condition 2)	Common
	b5	PLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Limit Alarm	User (condition 2)	Common
	b6	PHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High Limit Alarm	User (condition 2)	Common
	b7	LLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Low Limit Alarm	User (condition 2)	Common
	b8	HHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High High Limit Alarm	User (condition 2)	Common
	b9	SEA	BOOL	FALSE	TRUE	FALSE	—	—	Sensor Error Alarm	User (condition 2)	Common
	bA	OOA	BOOL	FALSE	TRUE	FALSE	—	—	Output Open Alarm	User (condition 2)	Common
	bB	DMLA	BOOL	FALSE	TRUE	FALSE	—	—	Output Variation Rate Limit Alarm	User (condition 2)	Common
	bE	SPA	BOOL	FALSE	TRUE	FALSE	—	—	Stop Alarm	User (condition 2)	Common
+4	b0	MLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output Low Limit Alarm	User	Common
	b1	MHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output High Limit Alarm	User	Common
	b2	DVLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Large Deviation Alarm	User	Common
	b3	DPNI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Negative Variation Rate Alarm	User	Common
	b4	DPPI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Positive Variation Rate Alarm	User	Common
	b5	PLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Limit Alarm	User	Common
	b6	PHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High Limit Alarm	User	Common
	b7	LLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Low Limit Alarm	User	Common
	b8	HHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High High Limit Alarm	User	Common
	b9	SEI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Sensor Error Alarm	User	Common
	bB	DMLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output Variation Rate Limit Alarm	User	Common
	bD	TRKF	BOOL	FALSE	TRUE	FALSE	—	—	Tracking Flag	User	Common
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User	Common

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+5	b0	MLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Low Limit Alarm	User	—
	b1	MHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output High Limit Alarm	User	—
	b2	DVLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Large Deviation Alarm	User	—
	b3	DPNL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Negative Variation Rate Alarm	User	—
	b4	DPPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Positive Variation Rate Alarm	User	—
	b5	PLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Limit Alarm	User	—
	b6	PHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High Limit Alarm	User	—
	b7	LLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Low Limit Alarm	User	—
	b8	HHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High High Limit Alarm	User	—
	b9	SENL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Sensor Error Alarm	User	—
	bA	OOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Open Alarm	User	—
	bB	DMLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Variation Rate Limit Alarm	User	—
bE	SPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Stop Alarm	User	—	
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—	
+8	UNIT	INT	0	127	0	—	—	Unit	User	—	
+9	N_	INT	0	4	1	—	—	No. of digits after the decimal point	User	—	
+10	PV	REAL	RL	RH	0.0	N	UNIT	Process Variable	System	M+P_PHPL	
+11											
+12	MV	REAL	-10	110	0.0	1	%	Manipulated Variable	User (condition 3)	M+P_OUT1 M+P_DUTY	
+13											
+14	SV	REAL	RL	RH	0.0	N	UNIT	Setting value	User	M+P_SPI(_T)	
+15											
+16	DV	REAL	-110	110	0.0	1	%	Deviation	System	M+P_SPI(_T)	
+17											
+18	MH	REAL	-10	110	100.0	1	%	MV High Limit Value	User	M+P_OUT1	
+19											
+20	ML	REAL	-10	110	0.0	1	%	MV Low Limit Value	User	M+P_OUT1	
+21											
+22	RH	REAL	-999999	999999	100.0	N	UNIT	PV Engineering Value High Limit	User	M+P_PHPL	
+23											
+24	RL	REAL	-999999	999999	0.0	N	UNIT	PV Engineering Value Low Limit	User	M+P_PHPL	
+25											
+26	PH_	REAL	RL	RH	100.0	N	UNIT	PV High Limit Alarm Value	User	M+P_PHPL	
+27											
+28	PL	REAL	RL	RH	0.0	N	UNIT	PV Low Limit Alarm Value	User	M+P_PHPL	
+29											

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+30	HH	REAL	RL	RH	100.0	N	UNIT	PV High High Limit Alarm Value	User	M+P_PHPL	
+31											
+32	LL	REAL	RL	RH	0.0	N	UNIT	PV Low Low Limit Alarm Value	User	M+P_PHPL	
+33											
+34	SH	REAL	RL	RH	100.0	N	UNIT	SV High Limit Value	User	—	
+35											
+36	SL	REAL	RL	RH	0.0	N	UNIT	SV Low Limit Value	User	—	
+37											
+38	ALPHA	REAL	0	1	0.2	2	—	PV Filter Coefficient	User	M+P_IN	
+39											
+40	HS	REAL	0	100	0.0	1	%	PV High/Low Limit Alarm Hysteresis	User	M+P_PHPL	
+41											
+42	CTIM	REAL	0	9999	0.0	2	s	Variation Rate Alarm Check Time	User	M+P_PHPL	
+43											
+44	DPL	REAL	0	100	100.0	1	%	Variation Rate Alarm Value	User	M+P_PHPL	
+45											
+46	ST_	REAL	0	9999	0.0	2	s	Operating Time	User	M+P_SPI(_T)	
+47											
+48	DML	REAL	0	100	100.0	1	%	Output Variation Rate High Limit Value	User	M+P_OUT1 M+P_DUTY	
+49											
+50	DVL	REAL	0	100	100.0	1	%	Deviation Limit Value	User	M+P_SPI(_T)	
+51											
+52	PROPORTIONAL	REAL	0	999	1.0	2	—	Gain	User	M+P_SPI(_T)	
+53											
+54	INTEGRAL	REAL	0	9999	10.0	1	s	Integral Time	User	M+P_SPI(_T)	
+55											
+56	STHT	REAL	0	9999	0.0	1	s	Sampling Period	User	M+P_SPI(_T)	
+57											
+58	GW	REAL	0	100	0.0	1	%	Gap Width	User	M+P_SPI(_T)	
+59											
+60	GG	REAL	0	99	1.0	1	—	Gap Gain	User	M+P_SPI(_T)	
+61											
+62	MVP	REAL	-999999	999999	0.0	1	—	MV Internal Operation Value	System	—	
+63											
+68	CTDUTY	REAL	0	9999	1.0	2	s	Control Output Cycle	User	M+P_DUTY	
+69											
+94	bE	OVR	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: OVERRIDE	System	—
	bF	SIM	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: SIMULATION	System	—

# IPD

Structure name: M+TM\_IPD

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+0	FUNC	INT	4	4	4	—	—	Tag Function Code	System	—
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG M+P_IPD(_T) M+P_OUT1 M+P_DUTY
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG M+P_IPD(_T) M+P_OUT1 M+P_DUTY
	b5	CAS	BOOL	FALSE	TRUE	FALSE	—	Control Mode: CASCADE	User (condition 1)	M+P_MCHG M+P_IPD(_T) M+P_OUT1 M+P_DUTY
	b9	CMV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER MV	User (condition 1)	M+P_MCHG M+P_IPD(_T) M+P_OUT1 M+P_DUTY
	bA	CSV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER SV	User (condition 1)	M+P_MCHG M+P_IPD(_T) M+P_OUT1 M+P_DUTY
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable MANUAL	User	M+P_MCHG
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable AUTO	User	M+P_MCHG
	b5	CASI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable CASCADE	User	M+P_MCHG
	b9	CMVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER MV	User	M+P_MCHG
	bA	CSVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER SV	User	M+P_MCHG
	bE	OVRI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable OVERRIDE	User	M+P_MCHG
	bF	SIMI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable SIMULATION	User	M+P_MCHG



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+3	b0	MLA	BOOL	FALSE	TRUE	FALSE	—	—	Output Low Limit Alarm	User (condition 2)	Common
	b1	MHA	BOOL	FALSE	TRUE	FALSE	—	—	Output High Limit Alarm	User (condition 2)	Common
	b2	DVLA	BOOL	FALSE	TRUE	FALSE	—	—	Large Deviation Alarm	User (condition 2)	Common
	b3	DPNA	BOOL	FALSE	TRUE	FALSE	—	—	Negative Variation Rate Alarm	User (condition 2)	Common
	b4	DPPA	BOOL	FALSE	TRUE	FALSE	—	—	Positive Variation Rate Alarm	User (condition 2)	Common
	b5	PLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Limit Alarm	User (condition 2)	Common
	b6	PHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High Limit Alarm	User (condition 2)	Common
	b7	LLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Low Limit Alarm	User (condition 2)	Common
	b8	HHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High High Limit Alarm	User (condition 2)	Common
	b9	SEA	BOOL	FALSE	TRUE	FALSE	—	—	Sensor Error Alarm	User (condition 2)	Common
	bA	OOA	BOOL	FALSE	TRUE	FALSE	—	—	Output Open Alarm	User (condition 2)	Common
	bB	DMLA	BOOL	FALSE	TRUE	FALSE	—	—	Output Variation Rate Limit Alarm	User (condition 2)	Common
	bE	SPA	BOOL	FALSE	TRUE	FALSE	—	—	Stop Alarm	User (condition 2)	Common
+4	b0	MLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output Low Limit Alarm	User	Common
	b1	MHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output High Limit Alarm	User	Common
	b2	DVLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Large Deviation Alarm	User	Common
	b3	DPNI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Negative Variation Rate Alarm	User	Common
	b4	DPPI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Positive Variation Rate Alarm	User	Common
	b5	PLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Limit Alarm	User	Common
	b6	PHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High Limit Alarm	User	Common
	b7	LLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Low Limit Alarm	User	Common
	b8	HHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High High Limit Alarm	User	Common
	b9	SEI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Sensor Error Alarm	User	Common
	bB	DMLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output Variation Rate Limit Alarm	User	Common
	bD	TRKF	BOOL	FALSE	TRUE	FALSE	—	—	Tracking Flag	User	Common
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User	Common



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+5	b0	MLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Low Limit Alarm	User	—
	b1	MHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output High Limit Alarm	User	—
	b2	DVLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Large Deviation Alarm	User	—
	b3	DPNL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Negative Variation Rate Alarm	User	—
	b4	DPPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Positive Variation Rate Alarm	User	—
	b5	PLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Limit Alarm	User	—
	b6	PHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High Limit Alarm	User	—
	b7	LLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Low Limit Alarm	User	—
	b8	HHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High High Limit Alarm	User	—
	b9	SENL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Sensor Error Alarm	User	—
	bA	OOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Open Alarm	User	—
	bB	DMLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Variation Rate Limit Alarm	User	—
bE	SPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Stop Alarm	User	—	
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—	
+8	UNIT	INT	0	127	0	—	—	Unit	User	—	
+9	N_	INT	0	4	1	—	—	No. of digits after the decimal point	User	—	
+10	PV	REAL	RL	RH	0.0	N	UNIT	Process Variable	System	M+P_PHPL	
+11											
+12	MV	REAL	-10	110	0.0	1	%	Manipulated Variable	User (condition 3)	M+P_OUT1 M+P_DUTY	
+13											
+14	SV	REAL	RL	RH	0.0	N	UNIT	Setting value	User	M+P_IPD(_T)	
+15											
+16	DV	REAL	-110	110	0.0	1	%	Deviation	System	M+P_IPD(_T)	
+17											
+18	MH	REAL	-10	110	100.0	1	%	MV High Limit Value	User	M+P_OUT1	
+19											
+20	ML	REAL	-10	110	0.0	1	%	MV Low Limit Value	User	M+P_OUT1	
+21											
+22	RH	REAL	-999999	999999	100.0	N	UNIT	PV Engineering Value High Limit	User	M+P_PHPL	
+23											
+24	RL	REAL	-999999	999999	0.0	N	UNIT	PV Engineering Value Low Limit	User	M+P_PHPL	
+25											
+26	PH_	REAL	RL	RH	100.0	N	UNIT	PV High Limit Alarm Value	User	M+P_PHPL	
+27											
+28	PL	REAL	RL	RH	0.0	N	UNIT	PV Low Limit Alarm Value	User	M+P_PHPL	
+29											



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+30	HH	REAL	RL	RH	100.0	N	UNIT	PV High High Limit Alarm Value	User	M+P_PHPL	
+31											
+32	LL	REAL	RL	RH	0.0	N	UNIT	PV Low Low Limit Alarm Value	User	M+P_PHPL	
+33											
+34	SH	REAL	RL	RH	100.0	N	UNIT	SV High Limit Value	User	—	
+35											
+36	SL	REAL	RL	RH	0.0	N	UNIT	SV Low Limit Value	User	—	
+37											
+38	ALPHA	REAL	0	1	0.2	2	—	PV Filter Coefficient	User	M+P_IN	
+39											
+40	HS	REAL	0	100	0.0	1	%	PV High/Low Limit Alarm Hysteresis	User	M+P_PHPL	
+41											
+42	CTIM	REAL	0	9999	0.0	2	s	Variation Rate Alarm Check Time	User	M+P_PHPL	
+43											
+44	DPL	REAL	0	100	100.0	1	%	Variation Rate Alarm Value	User	M+P_PHPL	
+45											
+46	CT	REAL	0	9999	1.0	2	s	Control Cycle	User	M+P_IPD(_T)	
+47											
+48	DML	REAL	0	100	100.0	1	%	Output Variation Rate High Limit Value	User	M+P_OUT1 M+P_DUTY	
+49											
+50	DVL	REAL	0	100	100.0	1	%	Deviation Limit Value	User	M+P_IPD(_T)	
+51											
+52	PROPORTIONAL	REAL	0	999	1.0	2	—	Gain	User	M+P_IPD(_T)	
+53											
+54	INTEGRAL	REAL	0	9999	10.0	1	s	Integral Time	User	M+P_IPD(_T)	
+55											
+56	DERIVATIVE	REAL	0	9999	0.0	1	s	Derivative Time	User	M+P_IPD(_T)	
+57											
+58	GW	REAL	0	100	0.0	1	%	Gap Width	User	M+P_IPD(_T)	
+59											
+60	GG	REAL	0	99	1.0	1	—	Gap Gain	User	M+P_IPD(_T)	
+61											
+62	MVP	REAL	-999999	999999	0.0	1	—	MV Internal Operation Value	System	—	
+63											
+68	CTDUTY	REAL	0	9999	1.0	2	s	Control Output Cycle	User	M+P_DUTY	
+69											
+94	bE	OVR	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: OVERRIDE	System	—
	bF	SIM	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: SIMULATION	System	—

# BPI

Structure name: M+TM\_BPI

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+0	FUNC	INT	5	5	5	—	—	Tag Function Code	System	—
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG M+P_BPI(_T) M+P_OUT1 M+P_DUTY
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG M+P_BPI(_T) M+P_OUT1 M+P_DUTY
	b5	CAS	BOOL	FALSE	TRUE	FALSE	—	Control Mode: CASCADE	User (condition 1)	M+P_MCHG M+P_BPI(_T) M+P_OUT1 M+P_DUTY
	b9	CMV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER MV	User (condition 1)	M+P_MCHG M+P_BPI(_T) M+P_OUT1 M+P_DUTY
	bA	CSV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER SV	User (condition 1)	M+P_MCHG M+P_BPI(_T) M+P_OUT1 M+P_DUTY
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable MANUAL	User	M+P_MCHG
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable AUTO	User	M+P_MCHG
	b5	CASI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable CASCADE	User	M+P_MCHG
	b9	CMVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER MV	User	M+P_MCHG
	bA	CSVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER SV	User	M+P_MCHG
	bE	OVRI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable OVERRIDE	User	M+P_MCHG
	bF	SIMI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable SIMULATION	User	M+P_MCHG



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+3	b0	MLA	BOOL	FALSE	TRUE	FALSE	—	—	Output Low Limit Alarm	User (condition 2)	Common
	b1	MHA	BOOL	FALSE	TRUE	FALSE	—	—	Output High Limit Alarm	User (condition 2)	Common
	b2	DVLA	BOOL	FALSE	TRUE	FALSE	—	—	Large Deviation Alarm	User (condition 2)	Common
	b3	DPNA	BOOL	FALSE	TRUE	FALSE	—	—	Negative Variation Rate Alarm	User (condition 2)	Common
	b4	DPPA	BOOL	FALSE	TRUE	FALSE	—	—	Positive Variation Rate Alarm	User (condition 2)	Common
	b5	PLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Limit Alarm	User (condition 2)	Common
	b6	PHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High Limit Alarm	User (condition 2)	Common
	b7	LLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Low Limit Alarm	User (condition 2)	Common
	b8	HHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High High Limit Alarm	User (condition 2)	Common
	b9	SEA	BOOL	FALSE	TRUE	FALSE	—	—	Sensor Error Alarm	User (condition 2)	Common
	bA	OOA	BOOL	FALSE	TRUE	FALSE	—	—	Output Open Alarm	User (condition 2)	Common
	bB	DMLA	BOOL	FALSE	TRUE	FALSE	—	—	Output Variation Rate Limit Alarm	User (condition 2)	Common
	bE	SPA	BOOL	FALSE	TRUE	FALSE	—	—	Stop Alarm	User (condition 2)	Common
+4	b0	MLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output Low Limit Alarm	User	Common
	b1	MHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output High Limit Alarm	User	Common
	b2	DVLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Large Deviation Alarm	User	Common
	b3	DPNI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Negative Variation Rate Alarm	User	Common
	b4	DPPI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Positive Variation Rate Alarm	User	Common
	b5	PLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Limit Alarm	User	Common
	b6	PHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High Limit Alarm	User	Common
	b7	LLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Low Limit Alarm	User	Common
	b8	HHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High High Limit Alarm	User	Common
	b9	SEI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Sensor Error Alarm	User	Common
	bB	DMLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output Variation Rate Limit Alarm	User	Common
	bD	TRKF	BOOL	FALSE	TRUE	FALSE	—	—	Tracking Flag	User	Common
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User	Common

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+5	b0	MLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Low Limit Alarm	User	—
	b1	MHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output High Limit Alarm	User	—
	b2	DVLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Large Deviation Alarm	User	—
	b3	DPNL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Negative Variation Rate Alarm	User	—
	b4	DPPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Positive Variation Rate Alarm	User	—
	b5	PLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Limit Alarm	User	—
	b6	PHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High Limit Alarm	User	—
	b7	LLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Low Limit Alarm	User	—
	b8	HHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High High Limit Alarm	User	—
	b9	SENL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Sensor Error Alarm	User	—
	bA	OOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Open Alarm	User	—
	bB	DMLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Variation Rate Limit Alarm	User	—
bE	SPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Stop Alarm	User	—	
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—	
+8	UNIT	INT	0	127	0	—	—	Unit	User	—	
+9	N_	INT	0	4	1	—	—	No. of digits after the decimal point	User	—	
+10	PV	REAL	RL	RH	0.0	N	UNIT	Process Variable	System	M+P_PHPL	
+11											
+12	MV	REAL	-10	110	0.0	1	%	Manipulated Variable	User (condition 3)	M+P_OUT1 M+P_DUTY	
+13											
+14	SV	REAL	RL	RH	0.0	N	UNIT	Setting value	User	M+P_BPI(_T)	
+15											
+16	DV	REAL	-110	110	0.0	1	%	Deviation	System	M+P_BPI(_T)	
+17											
+18	MH	REAL	-10	110	100.0	1	%	MV High Limit Value	User	M+P_OUT1	
+19											
+20	ML	REAL	-10	110	0.0	1	%	MV Low Limit Value	User	M+P_OUT1	
+21											
+22	RH	REAL	-999999	999999	100.0	N	UNIT	PV Engineering Value High Limit	User	M+P_PHPL	
+23											
+24	RL	REAL	-999999	999999	0.0	N	UNIT	PV Engineering Value Low Limit	User	M+P_PHPL	
+25											
+26	PH_	REAL	RL	RH	100.0	N	UNIT	PV High Limit Alarm Value	User	M+P_PHPL	
+27											
+28	PL	REAL	RL	RH	0.0	N	UNIT	PV Low Limit Alarm Value	User	M+P_PHPL	
+29											



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+30	HH	REAL	RL	RH	100.0	N	UNIT	PV High High Limit Alarm Value	User	M+P_PHPL	
+31											
+32	LL	REAL	RL	RH	0.0	N	UNIT	PV Low Low Limit Alarm Value	User	M+P_PHPL	
+33											
+34	SH	REAL	RL	RH	100.0	N	UNIT	SV High Limit Value	User	—	
+35											
+36	SL	REAL	RL	RH	0.0	N	UNIT	SV Low Limit Value	User	—	
+37											
+38	ALPHA	REAL	0	1	0.2	2	—	PV Filter Coefficient	User	M+P_IN	
+39											
+40	HS	REAL	0	100	0.0	1	%	PV High/Low Limit Alarm Hysteresis	User	M+P_PHPL	
+41											
+42	CTIM	REAL	0	9999	0.0	2	s	Variation Rate Alarm Check Time	User	M+P_PHPL	
+43											
+44	DPL	REAL	0	100	100.0	1	%	Variation Rate Alarm Value	User	M+P_PHPL	
+45											
+46	CT	REAL	0	9999	1.0	2	s	Control Cycle	User	M+P_BPI(_T)	
+47											
+48	DML	REAL	0	100	100.0	1	%	Output Variation Rate High Limit Value	User	M+P_OUT1 M+P_DUTY	
+49											
+50	DVL	REAL	0	100	100.0	1	%	Deviation Limit Value	User	M+P_BPI(_T)	
+51											
+52	PROPORTIONAL	REAL	0	999	1.0	2	—	Gain	User	M+P_BPI(_T)	
+53											
+54	INTEGRAL	REAL	0	9999	10.0	1	s	Integral Time	User	M+P_BPI(_T)	
+55											
+56	SDV	REAL	-999999	999999	0.0	1	—	DV Cumulative Value	System	M+P_BPI(_T)	
+57											
+58	GW	REAL	0	100	0.0	1	%	Gap Width	User	M+P_BPI(_T)	
+59											
+60	GG	REAL	0	99	1.0	1	—	Gap Gain	User	M+P_BPI(_T)	
+61											
+62	MVP	REAL	-999999	999999	0.0	1	—	MV Internal Operation Value	System	—	
+63											
+68	CTDUTY	REAL	0	9999	1.0	2	s	Control Output Cycle	User	M+P_DUTY	
+69											
+94	bE	OVR	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: OVERRIDE	System	—
	bF	SIM	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: SIMULATION	System	—

# R

Structure name: M+TM\_R

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+0	FUNC	INT	6	6	6	—	—	Tag Function Code	System	—
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG M+P_R(_T) M+P_OUT2
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG M+P_R(_T) M+P_OUT2
	b5	CAS	BOOL	FALSE	TRUE	FALSE	—	Control Mode: CASCADE	User (condition 1)	M+P_MCHG M+P_R(_T) M+P_OUT2
	b9	CMV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER MV	User (condition 1)	M+P_MCHG M+P_R(_T) M+P_OUT2
	bA	CSV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER SV	User (condition 1)	M+P_MCHG M+P_R(_T) M+P_OUT2
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable MANUAL	User	M+P_MCHG
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable AUTO	User	M+P_MCHG
	b5	CASI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable CASCADE	User	M+P_MCHG
	b9	CMVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER MV	User	M+P_MCHG
	bA	CSVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER SV	User	M+P_MCHG
	bE	OVRI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable OVERRIDE	User	M+P_MCHG
	bF	SIMI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable SIMULATION	User	M+P_MCHG



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+3	b0	MLA	BOOL	FALSE	TRUE	FALSE	—	—	Output Low Limit Alarm	User (condition 2)	Common
	b1	MHA	BOOL	FALSE	TRUE	FALSE	—	—	Output High Limit Alarm	User (condition 2)	Common
	b3	DPNA	BOOL	FALSE	TRUE	FALSE	—	—	Negative Variation Rate Alarm	User (condition 2)	Common
	b4	DPPA	BOOL	FALSE	TRUE	FALSE	—	—	Positive Variation Rate Alarm	User (condition 2)	Common
	b5	PLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Limit Alarm	User (condition 2)	Common
	b6	PHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High Limit Alarm	User (condition 2)	Common
	b7	LLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Low Limit Alarm	User (condition 2)	Common
	b8	HHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High High Limit Alarm	User (condition 2)	Common
	b9	SEA	BOOL	FALSE	TRUE	FALSE	—	—	Sensor Error Alarm	User (condition 2)	Common
	bA	OOA	BOOL	FALSE	TRUE	FALSE	—	—	Output Open Alarm	User (condition 2)	Common
	bB	DMLA	BOOL	FALSE	TRUE	FALSE	—	—	Output Variation Rate Limit Alarm	User (condition 2)	Common
	bE	SPA	BOOL	FALSE	TRUE	FALSE	—	—	Stop Alarm	User (condition 2)	Common
+4	b0	MLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output Low Limit Alarm	User	Common
	b1	MHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output High Limit Alarm	User	Common
	b3	DPNI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Negative Variation Rate Alarm	User	Common
	b4	DPPI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Positive Variation Rate Alarm	User	Common
	b5	PLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Limit Alarm	User	Common
	b6	PHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High Limit Alarm	User	Common
	b7	LLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Low Limit Alarm	User	Common
	b8	HHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High High Limit Alarm	User	Common
	b9	SEI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Sensor Error Alarm	User	Common
	bB	DMLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Output Variation Rate Limit Alarm	User	Common
	bD	TRKF	BOOL	FALSE	TRUE	FALSE	—	—	Tracking Flag	User	Common
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User	Common



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+5	b0	MLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Low Limit Alarm	User	—
	b1	MHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output High Limit Alarm	User	—
	b3	DPNL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Negative Variation Rate Alarm	User	—
	b4	DPPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Positive Variation Rate Alarm	User	—
	b5	PLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Limit Alarm	User	—
	b6	PHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High Limit Alarm	User	—
	b7	LLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Low Limit Alarm	User	—
	b8	HHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High High Limit Alarm	User	—
	b9	SENL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Sensor Error Alarm	User	—
	bA	OOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Open Alarm	User	—
	bB	DMLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Variation Rate Limit Alarm	User	—
bE	SPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Stop Alarm	User	—	
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—	
+8	UNIT	INT	0	127	0	—	—	Unit	User	—	
+9	N_	INT	0	4	1	—	—	No. of digits after the decimal point	User	—	
+10	PV	REAL	RL	RH	0.0	N	UNIT	Process Variable	System	M+P_PHPL	
+11											
+12	MV	REAL	-10	110	0.0	1	%	Manipulated Variable	User (condition 3)	M+P_OUT2	
+13											
+14	SV	REAL	RMIN	RMAX	0.0	1	%	Setting Value (SPR)	User	M+P_R(_T)	
+15											
+16	BIAS	REAL	-999999	999999	0.0	1	—	Bias	User	M+P_R(_T)	
+17											
+18	MH	REAL	-10	110	100.0	1	%	MV High Limit Value	User	M+P_OUT2	
+19											
+20	ML	REAL	-10	110	0.0	1	%	MV Low Limit Value	User	M+P_OUT2	
+21											
+22	RH	REAL	-999999	999999	100.0	N	UNIT	PV Engineering Value High Limit	User	M+P_PHPL	
+23											
+24	RL	REAL	-999999	999999	0.0	N	UNIT	PV Engineering Value Low Limit	User	M+P_PHPL	
+25											
+26	PH_	REAL	RL	RH	100.0	N	UNIT	PV High Limit Alarm Value	User	M+P_PHPL	
+27											
+28	PL	REAL	RL	RH	0.0	N	UNIT	PV Low Limit Alarm Value	User	M+P_PHPL	
+29											
+30	HH	REAL	RL	RH	100.0	N	UNIT	PV High High Limit Alarm Value	User	M+P_PHPL	
+31											



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+32	LL	REAL	RL	RH	0.0	N	UNIT	PV Low Low Limit Alarm Value	User	M+P_PHPL	
+33											
+34	SH	REAL	RMIN	RMAX	100.0	N	%	SV High Limit Value	User	—	
+35											
+36	SL	REAL	RMIN	RMAX	0.0	N	%	SV Low Limit Value	User	—	
+37											
+38	ALPHA	REAL	0	1	0.2	2	—	PV Filter Coefficient	User	M+P_IN	
+39											
+40	HS	REAL	0	100	0.0	1	%	PV High/Low Limit Alarm Hysteresis	User	M+P_PHPL	
+41											
+42	CTIM	REAL	0	9999	0.0	2	s	Variation Rate Alarm Check Time	User	M+P_PHPL	
+43											
+44	DPL	REAL	0	100	100.0	1	%	Variation Rate Alarm Value	User	M+P_PHPL	
+45											
+46	CT	REAL	0	9999	1.0	2	s	Control Cycle	User	M+P_R(_T)	
+47											
+48	DML	REAL	0	100	100.0	1	%	Output Variation Rate High Limit Value	User	M+P_OUT2	
+49											
+50	DR	REAL	0	999999	100.0	1	%	Variation Rate Limit Value	User	M+P_R(_T)	
+51											
+52	RMAX	REAL	0	999999	100.0	1	%	Ratio High Limit	User	M+P_R(_T)	
+53											
+54	RMIN	REAL	0	999999	0.0	1	%	Ratio Low Limit	User	M+P_R(_T)	
+55											
+56	RN	REAL	0	999999	0.0	1	%	Ratio Current Value	System	M+P_R(_T)	
+57											
+94	bE	OVR	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: OVERRIDE	System	—
	bF	SIM	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: SIMULATION	System	—

# ONF2

Structure name: M+TM\_ONF2

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+0	FUNC	INT	7	7	7	—	—	Tag Function Code	System	—
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG M+P_ONF2(_T)
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG M+P_ONF2(_T)
	b5	CAS	BOOL	FALSE	TRUE	FALSE	—	Control Mode: CASCADE	User (condition 1)	M+P_MCHG M+P_ONF2(_T)
	b9	CMV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER MV	User (condition 1)	M+P_MCHG M+P_ONF2(_T)
	bA	CSV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER SV	User (condition 1)	M+P_MCHG M+P_ONF2(_T)
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable MANUAL	User	M+P_MCHG
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable AUTO	User	M+P_MCHG
	b5	CASI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable CASCADE	User	M+P_MCHG
	b9	CMVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER MV	User	M+P_MCHG
	bA	CSVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER SV	User	M+P_MCHG
	bE	OVRI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable OVERRIDE	User	M+P_MCHG
+3	b3	DPNA	BOOL	FALSE	TRUE	FALSE	—	Negative Variation Rate Alarm	User (condition 2)	Common
	b4	DPPA	BOOL	FALSE	TRUE	FALSE	—	Positive Variation Rate Alarm	User (condition 2)	Common
	b5	PLA	BOOL	FALSE	TRUE	FALSE	—	Input Low Limit Alarm	User (condition 2)	Common
	b6	PHA	BOOL	FALSE	TRUE	FALSE	—	Input High Limit Alarm	User (condition 2)	Common
	b7	LLA	BOOL	FALSE	TRUE	FALSE	—	Input Low Low Limit Alarm	User (condition 2)	Common
	b8	HHA	BOOL	FALSE	TRUE	FALSE	—	Input High High Limit Alarm	User (condition 2)	Common
	b9	SEA	BOOL	FALSE	TRUE	FALSE	—	Sensor Error Alarm	User (condition 2)	Common
	bA	OOA	BOOL	FALSE	TRUE	FALSE	—	Output Open Alarm	User (condition 2)	Common
	bE	SPA	BOOL	FALSE	TRUE	FALSE	—	Stop Alarm	User (condition 2)	Common



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+4	b3	DPNI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Negative Variation Rate Alarm	User	Common
	b4	DPPI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Positive Variation Rate Alarm	User	Common
	b5	PLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Limit Alarm	User	Common
	b6	PHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High Limit Alarm	User	Common
	b7	LLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Low Limit Alarm	User	Common
	b8	HHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High High Limit Alarm	User	Common
	b9	SEI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Sensor Error Alarm	User	Common
	bD	TRKF	BOOL	FALSE	TRUE	FALSE	—	—	Tracking Flag	User	Common
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User	Common
+5	b3	DPNL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Negative Variation Rate Alarm	User	—
	b4	DPPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Positive Variation Rate Alarm	User	—
	b5	PLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Limit Alarm	User	—
	b6	PHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High Limit Alarm	User	—
	b7	LLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Low Limit Alarm	User	—
	b8	HHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High High Limit Alarm	User	—
	b9	SENL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Sensor Error Alarm	User	—
	bA	OOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Open Alarm	User	—
	bE	SPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Stop Alarm	User	—
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—	
+8	UNIT	INT	0	127	0	—	—	Unit	User	—	
+9	N_	INT	0	4	1	—	—	No. of digits after the decimal point	User	—	
+10	PV	REAL	RL	RH	0.0	N	UNIT	Process Variable	System	M+P_PHPL	
+11											
+12	MV	REAL	-10	110	0.0	1	%	Manipulated Variable	User (condition 3)	M+P_ONF2(_T)	
+13											
+14	SV	REAL	RL	RH	0.0	N	UNIT	Setting value	User	M+P_ONF2(_T)	
+15											
+16	DV	REAL	-110	110	0.0	1	%	Deviation	System	M+P_ONF2(_T)	
+17											
+18	HS0	REAL	0	100	0.0	1	%	Hysteresis	User	M+P_ONF2(_T)	
+19											
+22	RH	REAL	-999999	999999	100.0	N	UNIT	PV Engineering Value High Limit	User	M+P_PHPL	
+23											

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+24	RL	REAL	-999999	999999	0.0	N	UNIT	PV Engineering Value Low Limit	User	M+P_PHPL	
+25											
+26	PH_	REAL	RL	RH	100.0	N	UNIT	PV High Limit Alarm Value	User	M+P_PHPL	
+27											
+28	PL	REAL	RL	RH	0.0	N	UNIT	PV Low Limit Alarm Value	User	M+P_PHPL	
+29											
+30	HH	REAL	RL	RH	100.0	N	UNIT	PV High High Limit Alarm Value	User	M+P_PHPL	
+31											
+32	LL	REAL	RL	RH	0.0	N	UNIT	PV Low Low Limit Alarm Value	User	M+P_PHPL	
+33											
+34	SH	REAL	RL	RH	100.0	N	UNIT	SV High Limit Value	User	—	
+35											
+36	SL	REAL	RL	RH	0.0	N	UNIT	SV Low Limit Value	User	—	
+37											
+38	ALPHA	REAL	0	1	0.2	2	—	PV Filter Coefficient	User	M+P_IN	
+39											
+40	HS	REAL	0	100	0.0	1	%	PV High/Low Limit Alarm Hysteresis	User	M+P_PHPL	
+41											
+42	CTIM	REAL	0	9999	0.0	2	s	Variation Rate Alarm Check Time	User	M+P_PHPL	
+43											
+44	DPL	REAL	0	100	100.0	1	%	Variation Rate Alarm Value	User	M+P_PHPL	
+45											
+46	CT	REAL	0	9999	1.0	2	s	Control Cycle	User	M+P_ONF2(T)	
+47											
+94	bE	OVR	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: OVERRIDE	System	—



# ONF3

Structure name: M+TM\_ONF3

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+0	FUNC	INT	8	8	8	—	—	Tag Function Code	System	—
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG M+P_ONF3(_T)
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG M+P_ONF3(_T)
	b5	CAS	BOOL	FALSE	TRUE	FALSE	—	Control Mode: CASCADE	User (condition 1)	M+P_MCHG M+P_ONF3(_T)
	b9	CMV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER MV	User (condition 1)	M+P_MCHG M+P_ONF3(_T)
	bA	CSV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER SV	User (condition 1)	M+P_MCHG M+P_ONF3(_T)
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable MANUAL	User	M+P_MCHG
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable AUTO	User	M+P_MCHG
	b5	CASI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable CASCADE	User	M+P_MCHG
	b9	CMVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER MV	User	M+P_MCHG
	bA	CSVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER SV	User	M+P_MCHG
	bE	OVRI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable OVERRIDE	User	M+P_MCHG
+3	b3	DPNA	BOOL	FALSE	TRUE	FALSE	—	Negative Variation Rate Alarm	User (condition 2)	Common
	b4	DPPA	BOOL	FALSE	TRUE	FALSE	—	Positive Variation Rate Alarm	User (condition 2)	Common
	b5	PLA	BOOL	FALSE	TRUE	FALSE	—	Input Low Limit Alarm	User (condition 2)	Common
	b6	PHA	BOOL	FALSE	TRUE	FALSE	—	Input High Limit Alarm	User (condition 2)	Common
	b7	LLA	BOOL	FALSE	TRUE	FALSE	—	Input Low Low Limit Alarm	User (condition 2)	Common
	b8	HHA	BOOL	FALSE	TRUE	FALSE	—	Input High High Limit Alarm	User (condition 2)	Common
	b9	SEA	BOOL	FALSE	TRUE	FALSE	—	Sensor Error Alarm	User (condition 2)	Common
	bA	OOA	BOOL	FALSE	TRUE	FALSE	—	Output Open Alarm	User (condition 2)	Common
	bE	SPA	BOOL	FALSE	TRUE	FALSE	—	Stop Alarm	User (condition 2)	Common

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+4	b3	DPNI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Negative Variation Rate Alarm	User	Common
	b4	DPPI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Positive Variation Rate Alarm	User	Common
	b5	PLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Limit Alarm	User	Common
	b6	PHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High Limit Alarm	User	Common
	b7	LLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Low Limit Alarm	User	Common
	b8	HHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High High Limit Alarm	User	Common
	b9	SEI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Sensor Error Alarm	User	Common
	bD	TRKF	BOOL	FALSE	TRUE	FALSE	—	—	Tracking Flag	User	Common
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User	Common
+5	b3	DPNL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Negative Variation Rate Alarm	User	—
	b4	DPPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Positive Variation Rate Alarm	User	—
	b5	PLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Limit Alarm	User	—
	b6	PHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High Limit Alarm	User	—
	b7	LLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Low Limit Alarm	User	—
	b8	HHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High High Limit Alarm	User	—
	b9	SENL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Sensor Error Alarm	User	—
	bA	OOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Open Alarm	User	—
	bE	SPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Stop Alarm	User	—
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—	
+8	UNIT	INT	0	127	0	—	—	Unit	User	—	
+9	N_	INT	0	4	1	—	—	No. of digits after the decimal point	User	—	
+10	PV	REAL	RL	RH	0.0	N	UNIT	Process Variable	System	M+P_PHPL	
+11											
+12	MV	REAL	-10	110	0.0	1	%	Manipulated Variable	User (condition 3)	M+P_ONF3(_T)	
+13											
+14	SV	REAL	RL	RH	0.0	N	UNIT	Setting value	User	M+P_ONF3(_T)	
+15											
+16	DV	REAL	-110	110	0.0	1	%	Deviation	System	M+P_ONF3(_T)	
+17											
+18	HS0	REAL	0	100	0.0	1	%	Hysteresis	User	M+P_ONF3(_T)	
+19											
+20	HS1	REAL	0	100	0.0	1	%	Hysteresis	User	M+P_ONF3(_T)	
+21											

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+22	RH	REAL	-999999	999999	100.0	N	UNIT	PV Engineering Value High Limit	User	M+P_PHPL	
+23											
+24	RL	REAL	-999999	999999	0.0	N	UNIT	PV Engineering Value Low Limit	User	M+P_PHPL	
+25											
+26	PH_	REAL	RL	RH	100.0	N	UNIT	PV High Limit Alarm Value	User	M+P_PHPL	
+27											
+28	PL	REAL	RL	RH	0.0	N	UNIT	PV Low Limit Alarm Value	User	M+P_PHPL	
+29											
+30	HH	REAL	RL	RH	100.0	N	UNIT	PV High High Limit Alarm Value	User	M+P_PHPL	
+31											
+32	LL	REAL	RL	RH	0.0	N	UNIT	PV Low Low Limit Alarm Value	User	M+P_PHPL	
+33											
+34	SH	REAL	RL	RH	100.0	N	UNIT	SV High Limit Value	User	—	
+35											
+36	SL	REAL	RL	RH	0.0	N	UNIT	SV Low Limit Value	User	—	
+37											
+38	ALPHA	REAL	0	1	0.2	2	—	PV Filter Coefficient	User	M+P_IN	
+39											
+40	HS	REAL	0	100	0.0	1	%	PV High/Low Limit Alarm Hysteresis	User	M+P_PHPL	
+41											
+42	CTIM	REAL	0	9999	0.0	2	s	Variation Rate Alarm Check Time	User	M+P_PHPL	
+43											
+44	DPL	REAL	0	100	100.0	1	%	Variation Rate Alarm Value	User	M+P_PHPL	
+45											
+46	CT	REAL	0	9999	1.0	2	s	Control Cycle	User	M+P_ONF3(_T)	
+47											
+94	bE	OVR	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: OVERRIDE	System	—



# MONI

Structure name: M+TM\_MONI

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+0	FUNC	INT	11	11	11	—	—	Tag Function Code	System	—
+2	bE OVRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable OVERRIDE	User	—
+3	b3 DPNA	BOOL	FALSE	TRUE	FALSE	—	—	Negative Variation Rate Alarm	User (condition 2)	Common
	b4 DPPA	BOOL	FALSE	TRUE	FALSE	—	—	Positive Variation Rate Alarm	User (condition 2)	Common
	b5 PLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Limit Alarm	User (condition 2)	Common
	b6 PHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High Limit Alarm	User (condition 2)	Common
	b7 LLA	BOOL	FALSE	TRUE	FALSE	—	—	Input Low Low Limit Alarm	User (condition 2)	Common
	b8 HHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High High Limit Alarm	User (condition 2)	Common
	b9 SEA	BOOL	FALSE	TRUE	FALSE	—	—	Sensor Error Alarm	User (condition 2)	Common
	bE SPA	BOOL	FALSE	TRUE	FALSE	—	—	Stop Alarm	User (condition 2)	Common
+4	b3 DPNI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Negative Variation Rate Alarm	User	Common
	b4 DPPI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Positive Variation Rate Alarm	User	Common
	b5 PLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Limit Alarm	User	Common
	b6 PHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High Limit Alarm	User	Common
	b7 LLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Low Limit Alarm	User	Common
	b8 HHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High High Limit Alarm	User	Common
	b9 SEI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Sensor Error Alarm	User	Common
	bF ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User	Common
+5	b3 DPNL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Negative Variation Rate Alarm	User	—
	b4 DPPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Positive Variation Rate Alarm	User	—
	b5 PLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Limit Alarm	User	—
	b6 PHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High Limit Alarm	User	—
	b7 LLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Low Limit Alarm	User	—
	b8 HHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High High Limit Alarm	User	—
	b9 SENL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Sensor Error Alarm	User	—
	bE SPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Stop Alarm	User	—



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—	
+8	UNIT	INT	0	127	0	—	—	Unit	User	—	
+9	N_	INT	0	4	1	—	—	No. of digits after the decimal point	User	—	
+10	PV	REAL	RL	RH	0.0	N	UNIT	Process Variable	System	M+P_PHPL	
+11											
+22	RH	REAL	-999999	999999	100.0	N	UNIT	PV Engineering Value High Limit	User	M+P_PHPL	
+23											
+24	RL	REAL	-999999	999999	0.0	N	UNIT	PV Engineering Value Low Limit	User	M+P_PHPL	
+25											
+26	PH_	REAL	RL	RH	100.0	N	UNIT	PV High Limit Alarm Value	User	M+P_PHPL	
+27											
+28	PL	REAL	RL	RH	0.0	N	UNIT	PV Low Limit Alarm Value	User	M+P_PHPL	
+29											
+30	HH	REAL	RL	RH	100.0	N	UNIT	PV High High Limit Alarm Value	User	M+P_PHPL	
+31											
+32	LL	REAL	RL	RH	0.0	N	UNIT	PV Low Low Limit Alarm Value	User	M+P_PHPL	
+33											
+38	ALPHA	REAL	0	1	0.2	2	—	PV Filter Coefficient	User	M+P_IN	
+39											
+40	HS	REAL	0	100	0.0	1	%	PV High/Low Limit Alarm Hysteresis	User	M+P_PHPL	
+41											
+42	CTIM	REAL	0	9999	0.0	2	s	Variation Rate Alarm Check Time	User	M+P_PHPL	
+43											
+44	DPL	REAL	0	100	100.0	1	%	Variation Rate Alarm Value	User	M+P_PHPL	
+45											
+94	bE	OVR	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: OVERRIDE	System	—

# SWM

Structure name: M+TM\_SWM

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+0	FUNC	INT	19	19	19	—	—	Tag Function Code	System	—
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG
	b5	CAS	BOOL	FALSE	TRUE	FALSE	—	Control Mode: CASCADE	User (condition 1)	M+P_MCHG
	bA	CSV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER SV	User (condition 1)	M+P_MCHG
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable MANUAL	User	M+P_MCHG
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable AUTO	User	M+P_MCHG
	b5	CASI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable CASCADE	User	M+P_MCHG
	bA	CSVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER SV	User	M+P_MCHG
	bC	TSTPI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable TAG STOP	User	M+P_MCHG
	bE	OVRI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable OVERRIDE	User	M+P_MCHG
+3	b2	DVLA	BOOL	FALSE	TRUE	FALSE	—	Large Deviation Alarm	User (condition 2)	Common
	b3	DPNA	BOOL	FALSE	TRUE	FALSE	—	Negative Variation Rate Alarm	User (condition 2)	Common
	b4	DPPA	BOOL	FALSE	TRUE	FALSE	—	Positive Variation Rate Alarm	User (condition 2)	Common
	b5	PLA	BOOL	FALSE	TRUE	FALSE	—	Input Low Limit Alarm	User (condition 2)	Common
	b6	PHA	BOOL	FALSE	TRUE	FALSE	—	Input High Limit Alarm	User (condition 2)	Common
	b7	LLA	BOOL	FALSE	TRUE	FALSE	—	Input Low Low Limit Alarm	User (condition 2)	Common
	b8	HHA	BOOL	FALSE	TRUE	FALSE	—	Input High High Limit Alarm	User (condition 2)	Common
	b9	SEA	BOOL	FALSE	TRUE	FALSE	—	Sensor Error Alarm	User (condition 2)	Common
	bA	OOA	BOOL	FALSE	TRUE	FALSE	—	Output Open Alarm	User (condition 2)	Common
	bE	SPA	BOOL	FALSE	TRUE	FALSE	—	Stop Alarm	User (condition 2)	Common



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+4	b2	DVLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Large Deviation Alarm	User	Common
	b3	DPNI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Negative Variation Rate Alarm	User	Common
	b4	DPPI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Positive Variation Rate Alarm	User	Common
	b5	PLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Limit Alarm	User	Common
	b6	PHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High Limit Alarm	User	Common
	b7	LLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Low Limit Alarm	User	Common
	b8	HHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High High Limit Alarm	User	Common
	b9	SEI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Sensor Error Alarm	User	Common
	bD	TRKF	BOOL	FALSE	TRUE	FALSE	—	—	Tracking Flag	User	Common
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User	Common
+5	b2	DVLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Large Deviation Alarm	User	—
	b3	DPNL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Negative Variation Rate Alarm	User	—
	b4	DPPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Positive Variation Rate Alarm	User	—
	b5	PLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Limit Alarm	User	—
	b6	PHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High Limit Alarm	User	—
	b7	LLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Low Limit Alarm	User	—
	b8	HHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High High Limit Alarm	User	—
	b9	SENL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Sensor Error Alarm	User	—
	bA	OOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Open Alarm	User	—
	bE	SPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Stop Alarm	User	—
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—	
+8	UNIT	INT	0	127	0	—	—	Unit	User	—	
+9	N_	INT	0	4	1	—	—	No. of digits after the decimal point	User	—	
+10	PV	REAL	RL	RH	0.0	N	UNIT	Process Variable	System	M+P_PHPL	
+11											
+12	MV	REAL	-10	110	0.0	1	%	Manipulated Variable	User	M+P_MSET_	
+13											
+14	SVC	REAL	RL	RH	0.0	N	UNIT	Setting Value (Current)	System	M+P_MSET_	
+15											
+16	DV	REAL	-110	110	0.0	1	%	Deviation	System	M+P_MSET_	
+17											
+22	RH	REAL	-999999	999999	100.0	N	UNIT	PV Engineering Value High Limit	User	M+P_PHPL	
+23											

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+24	RL	REAL	-999999	999999	0.0	N	UNIT	PV Engineering Value Low Limit	User	M+P_PHPL	
+25											
+26	PH_	REAL	RL	RH	100.0	N	UNIT	PV High Limit Alarm Value	User	M+P_PHPL	
+27											
+28	PL	REAL	RL	RH	0.0	N	UNIT	PV Low Limit Alarm Value	User	M+P_PHPL	
+29											
+30	HH	REAL	RL	RH	100.0	N	UNIT	PV High High Limit Alarm Value	User	M+P_PHPL	
+31											
+32	LL	REAL	RL	RH	0.0	N	UNIT	PV Low Low Limit Alarm Value	User	M+P_PHPL	
+33											
+34	SH	REAL	RL	RH	100.0	N	UNIT	SV High Limit Value	User	M+P_MSET_	
+35											
+36	SL	REAL	RL	RH	0.0	N	UNIT	SV Low Limit Value	User	M+P_MSET_	
+37											
+38	ALPHA	REAL	0	1	0.2	2	—	PV Filter Coefficient	User	M+P_IN	
+39											
+40	HS	REAL	0	100	0.0	1	%	PV High/Low Limit Alarm Hysteresis	User	M+P_PHPL	
+41											
+42	CTIM	REAL	0	9999	0.0	2	s	Variation Rate Alarm Check Time	User	M+P_PHPL	
+43											
+44	DPL	REAL	0	100	100.0	1	%	Variation Rate Alarm Value	User	M+P_PHPL	
+45											
+46	CT	REAL	0	9999	1.0	2	s	Control Cycle	User	M+P_MSET_	
+47											
+50	DVL	REAL	0	100	100.0	1	%	Deviation Limit Value	User	M+P_MSET_	
+51											
+87	bD	SVLA	BOOL	FALSE	TRUE	FALSE	—	—	SV Low Alarm	System	Common
	bE	SVHA	BOOL	FALSE	TRUE	FALSE	—	—	SV High Alarm	System	Common
	bF	DSVLA	BOOL	FALSE	TRUE	FALSE	—	—	SV Variation Rate Limit Alarm	System	Common
+88	bD	SVLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable SV Low Alarm	User	Common
	bE	SVHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable SV High Alarm	User	Common
	bF	DSVLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable SV Variation Rate Limit Alarm	User	Common
+89	bD	SVLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of SV Low Alarm	User	Common
	bE	SVHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of SV High Alarm	User	Common
	bF	DSVLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of SV Variation Rate Limit Alarm	User	Common
+90	SV	REAL	RL	RH	0.0	N	UNIT	Setting Value (Target)	User	—	
+91											
+92	DSVL	REAL	0	1	100.0	1	%	SV Variation Rate High Limit Value	User	M+P_MSET_	
+93											
+94	bD	TSTP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: TAG STOP	System	—
	bE	OVR	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: OVERRIDE	System	—



# MWM

Structure name: M+TM\_MWM

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+0	FUNC	INT	12	12	12	—	—	Tag Function Code	System	—
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG M+P_MOUT
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG M+P_MOUT
	b5	CAS	BOOL	FALSE	TRUE	FALSE	—	Control Mode: CASCADE	User (condition 1)	M+P_MCHG M+P_MOUT
	b9	CMV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER MV	User (condition 1)	M+P_MCHG M+P_MOUT
	bA	CSV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER SV	User (condition 1)	M+P_MCHG M+P_MOUT
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable MANUAL	User	M+P_MCHG
	b4	AUTI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable AUTO	User	M+P_MCHG
	b5	CASI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable CASCADE	User	M+P_MCHG
	b9	CMVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER MV	User	M+P_MCHG
	bA	CSVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER SV	User	M+P_MCHG
	bE	OVRI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable OVERRIDE	User	M+P_MCHG
+3	b3	DPNA	BOOL	FALSE	TRUE	FALSE	—	Negative Variation Rate Alarm	User (condition 2)	Common
	b4	DPPA	BOOL	FALSE	TRUE	FALSE	—	Positive Variation Rate Alarm	User (condition 2)	Common
	b5	PLA	BOOL	FALSE	TRUE	FALSE	—	Input Low Limit Alarm	User (condition 2)	Common
	b6	PHA	BOOL	FALSE	TRUE	FALSE	—	Input High Limit Alarm	User (condition 2)	Common
	b7	LLA	BOOL	FALSE	TRUE	FALSE	—	Input Low Low Limit Alarm	User (condition 2)	Common
	b8	HHA	BOOL	FALSE	TRUE	FALSE	—	Input High High Limit Alarm	User (condition 2)	Common
	b9	SEA	BOOL	FALSE	TRUE	FALSE	—	Sensor Error Alarm	User (condition 2)	Common
	bA	OOA	BOOL	FALSE	TRUE	FALSE	—	Output Open Alarm	User (condition 2)	Common
	bE	SPA	BOOL	FALSE	TRUE	FALSE	—	Stop Alarm	User (condition 2)	Common

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+4	b3	DPNI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Negative Variation Rate Alarm	User	Common
	b4	DPPI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Positive Variation Rate Alarm	User	Common
	b5	PLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Limit Alarm	User	Common
	b6	PHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High Limit Alarm	User	Common
	b7	LLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Low Limit Alarm	User	Common
	b8	HHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High High Limit Alarm	User	Common
	b9	SEI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Sensor Error Alarm	User	Common
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User	Common
+5	b3	DPNL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Negative Variation Rate Alarm	User	—
	b4	DPPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Positive Variation Rate Alarm	User	—
	b5	PLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Limit Alarm	User	—
	b6	PHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High Limit Alarm	User	—
	b7	LLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Low Limit Alarm	User	—
	b8	HHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High High Limit Alarm	User	—
	b9	SENL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Sensor Error Alarm	User	—
	bA	OOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Open Alarm	User	—
	bE	SPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Stop Alarm	User	—
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—	
+8	UNIT	INT	0	127	0	—	—	Unit	User	—	
+9	N_	INT	0	4	1	—	—	No. of digits after the decimal point	User	—	
+10	PV	REAL	RL	RH	0.0	N	UNIT	Process Variable	System	M+P_PHPL	
+11											
+12	MV	REAL	-10	110	0.0	1	%	Manipulated Variable	User (condition 3)	M+P_MOUT	
+13											
+18	MH	REAL	-10	110	100.0	1	%	MV High Limit Value	User	M+P_MOUT	
+19											
+20	ML	REAL	-10	110	0.0	1	%	MV Low Limit Value	User	M+P_MOUT	
+21											
+22	RH	REAL	-999999	999999	100.0	N	UNIT	PV Engineering Value High Limit	User	M+P_PHPL	
+23											
+24	RL	REAL	-999999	999999	0.0	N	UNIT	PV Engineering Value Low Limit	User	M+P_PHPL	
+25											
+26	PH_	REAL	RL	RH	100.0	N	UNIT	PV High Limit Alarm Value	User	M+P_PHPL	
+27											



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+28	PL	REAL	RL	RH	0.0	N	UNIT	PV Low Limit Alarm Value	User	M+P_PHPL	
+29											
+30	HH	REAL	RL	RH	100.0	N	UNIT	PV High High Limit Alarm Value	User	M+P_PHPL	
+31											
+32	LL	REAL	RL	RH	0.0	N	UNIT	PV Low Low Limit Alarm Value	User	M+P_PHPL	
+33											
+38	ALPHA	REAL	0	1	0.2	2	—	PV Filter Coefficient	User	M+P_IN	
+39											
+40	HS	REAL	0	100	0.0	1	%	PV High/Low Limit Alarm Hysteresis	User	M+P_PHPL	
+41											
+42	CTIM	REAL	0	9999	0.0	2	s	Variation Rate Alarm Check Time	User	M+P_PHPL	
+43											
+44	DPL	REAL	0	100	100.0	1	%	Variation Rate Alarm Value	User	M+P_PHPL	
+45											
+94	bE	OVR	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: OVERRIDE	System	—



# BC

Structure name: M+TM\_BC

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+0	FUNC	INT	15	15	15	—	—	Tag Function Code	System	—
+3	b4 DPPA	BOOL	FALSE	TRUE	FALSE	—	—	Positive Variation Rate Alarm	User (condition 2)	Common
	b6 PHA	BOOL	FALSE	TRUE	FALSE	—	—	Input High Limit Alarm	User (condition 2)	Common
+4	b4 DPPI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Positive Variation Rate Alarm	User	Common
	b6 PHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High Limit Alarm	User	Common
	bF ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User	Common
+5	b4 DPPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Positive Variation Rate Alarm	User	—
	b6 PHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High Limit Alarm	User	—
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—
+8	UNIT	INT	0	127	0	—	—	Unit	User	—
+10	PV	DINT	0	99999999	0	—	UNIT	Process Variable (Integer part)	System	M+P_PSUM
+11										
+12	SUM2	DINT	0	999	0	—	—	Process Variable(Decimal Part)	System	M+P_PSUM
+13										
+14	SV1	DINT	0	99999999	0	—	UNIT	Setting Value 1 (Preset)	User	M+P_BC
+15										
+16	SV2	DINT	0	99999999	0	—	UNIT	Setting Value 2 (Preset)	User	M+P_BC
+17										
+18	SV	DINT	0	99999999	0	—	UNIT	Setting value	User	—
+19										
+26	PH_	DINT	0	99999999	0	—	UNIT	PV High Limit Alarm Value	User	M+P_BC
+27										
+42	CTIM	REAL	0	9999	0.0	2	s	Variation Rate Alarm Check Time	User	M+P_BC
+43										
+44	DPL	DINT	0	99999999	99999999	—	UNIT	Variation Rate Alarm Value	User	M+P_BC
+45										
+94	b0 DOM_RUN_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Run by PC	Tag data access control	—
	b1 DOM_HOLD_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Hold by PC	Tag data access control	—
	b2 DOM_STOP_RESET_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Stop/Reset by PC	Tag data access control	—
	b3 DOM_RESET_START_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Reset/Start by PC	Tag data access control	—



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+95	b0	DIM_RUN	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Run	System	—
	b1	DIM_HOLD	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Hold	System	—
	b2	DIM_STOP_RESET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Stop reset	System	—
	b5	DIM_PRE_COMP1	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Pre-complete 1	System	—
	b6	DIM_PRE_COMP2	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Pre-complete 2	System	—
	b7	DIM_COMP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Complete	System	—

# PSUM

Structure name: M+TM\_PSUM

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+0	FUNC	INT	16	16	16	—	—	Tag Function Code	System	—	
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—	
+8	UNIT	INT	0	127	0	—	—	Unit	User	—	
+10	PV	DINT	0	99999999	0	—	UNIT	Process Variable (Integer part)	System	M+P_PSUM	
+11											
+12	SUM2	DINT	0	999	0	—	—	Process Variable(Decimal Part)	System	M+P_PSUM	
+13											
+94	b0	DOM_RUN_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Run by PC	Tag data access control	—
	b1	DOM_HOLD_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Hold by PC	Tag data access control	—
	b2	DOM_STOP_RESET_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Stop/Reset by PC	Tag data access control	—
	b3	DOM_RESET_START_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Reset/Start by PC	Tag data access control	—
+95	b0	DIM_RUN	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Run	System	—
	b1	DIM_HOLD	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Hold	System	—
	b2	DIM_STOP_RESET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Stop reset	System	—



# SEL

Structure name: M+TM\_SEL

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+0	FUNC	INT	13	13	13	—	—	Tag Function Code	System	—
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG M+P_SEL(_T1) (_T2)(_T3_)
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG M+P_SEL(_T1) (_T2)(_T3_)
	b5	CAS	BOOL	FALSE	TRUE	FALSE	—	Control Mode: CASCADE	User (condition 1)	M+P_MCHG M+P_SEL(_T1) (_T2)(_T3_)
	b9	CMV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER MV	User (condition 1)	M+P_MCHG M+P_SEL(_T1) (_T2)(_T3_)
	bA	CSV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER SV	User (condition 1)	M+P_MCHG M+P_SEL(_T1) (_T2)(_T3_)
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable MANUAL	User	M+P_MCHG
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable AUTO	User	M+P_MCHG
	b5	CASI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable CASCADE	User	M+P_MCHG
	b9	CMVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER MV	User	M+P_MCHG
	bA	CSVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER SV	User	M+P_MCHG
+3	b0	MLA	BOOL	FALSE	TRUE	FALSE	—	Output Low Limit Alarm	User (condition 2)	Common
	b1	MHA	BOOL	FALSE	TRUE	FALSE	—	Output High Limit Alarm	User (condition 2)	Common
	bA	OOA	BOOL	FALSE	TRUE	FALSE	—	Output Open Alarm	User (condition 2)	Common
	bB	DMLA	BOOL	FALSE	TRUE	FALSE	—	Output Variation Rate Limit Alarm	User (condition 2)	Common
	bE	SPA	BOOL	FALSE	TRUE	FALSE	—	Stop Alarm	User (condition 2)	Common
+4	b0	MLI	BOOL	FALSE	TRUE	FALSE	—	Disable Output Low Limit Alarm	User	Common
	b1	MHI	BOOL	FALSE	TRUE	FALSE	—	Disable Output High Limit Alarm	User	Common
	bB	DMLI	BOOL	FALSE	TRUE	FALSE	—	Disable Output Variation Rate Limit Alarm	User	Common
	bD	TRKF	BOOL	FALSE	TRUE	FALSE	—	Tracking Flag	User	Common
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	Disable All Alarms	User	Common

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+5	b0	MLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Low Limit Alarm	User	—
	b1	MHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output High Limit Alarm	User	—
	bA	OOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Open Alarm	User	—
	bB	DMLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Variation Rate Limit Alarm	User	—
	bE	SPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Stop Alarm	User	—
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—	
+8	UNIT	INT	0	127	0	—	—	Unit	User	—	
+9	N_	INT	0	4	1	—	—	No. of digits after the decimal point	User	—	
+10	PV	REAL	RL	RH	0.0	N	UNIT	Selection Value	System	M+P_SEL(_T1) (_T2)(_T3_)	
+11											
+12	MV	REAL	-10	110	0.0	1	%	Manipulated Variable	User (condition 3)	M+P_SEL(_T1) (_T2)(_T3_)	
+13											
+14	PV1	REAL	RL	RH	0.0	N	UNIT	Process Variable 1	System	M+P_SEL(_T1) (_T2)(_T3_)	
+15											
+16	PV2	REAL	RL	RH	0.0	N	UNIT	Process Variable 2	System	M+P_SEL(_T1) (_T2)(_T3_)	
+17											
+18	MH	REAL	-10	110	100.0	1	%	MV High Limit Value	User	M+P_SEL(_T1) (_T2)(_T3_)	
+19											
+20	ML	REAL	-10	110	0.0	1	%	MV Low Limit Value	User	M+P_SEL(_T1) (_T2)(_T3_)	
+21											
+22	RH	REAL	-999999	999999	100.0	N	UNIT	PV Engineering Value High Limit	User	M+P_SEL(_T1) (_T2)(_T3_)	
+23											
+24	RL	REAL	-999999	999999	0.0	N	UNIT	PV Engineering Value Low Limit	User	M+P_SEL(_T1) (_T2)(_T3_)	
+25											
+26	SLNO	INT	1	2	0	—	—	Selection No.	System	M+P_SEL(_T1) (_T2)(_T3_)	
+46	CT	REAL	0	9999	1.0	2	s	Control Cycle	User	M+P_SEL_T3_	
+47											
+48	DML	REAL	0	100	100.0	1	%	Output Variation Rate High Limit Value	User	M+P_SEL(_T1) (_T2)(_T3_)	
+49											
+62	MVP	REAL	-999999	999999	0.0	1	—	MV Internal Operation Value	System	M+P_SEL_T3_	
+63											



# MOUT

Structure name: M+TM\_MOUT

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+0	FUNC	INT	10	10	10	—	—	Tag Function Code	System	—
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG M+P_MOUT
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG M+P_MOUT
	b5	CAS	BOOL	FALSE	TRUE	FALSE	—	Control Mode: CASCADE	User (condition 1)	M+P_MCHG M+P_MOUT
	b9	CMV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER MV	User (condition 1)	M+P_MCHG M+P_MOUT
	bA	CSV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER SV	User (condition 1)	M+P_MCHG M+P_MOUT
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable MANUAL	User	M+P_MCHG
	b4	AUTI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable AUTO	User	M+P_MCHG
	b5	CASI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable CASCADE	User	M+P_MCHG
	b9	CMVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER MV	User	M+P_MCHG
	bA	CSVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER SV	User	M+P_MCHG
+3	bA	OOA	BOOL	FALSE	TRUE	FALSE	—	Output Open Alarm	User (condition 2)	Common
	bE	SPA	BOOL	FALSE	TRUE	FALSE	—	Stop Alarm	User (condition 2)	Common
+5	bA	OOL	BOOL	FALSE	TRUE	FALSE	—	Alarm Level of Output Open Alarm	User	—
	bE	SPL	BOOL	FALSE	TRUE	FALSE	—	Alarm Level of Stop Alarm	User	—
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—
+8	UNIT	INT	0	127	0	—	—	Unit	User	—
+9	N_	INT	0	4	1	—	—	No. of digits after the decimal point	User	—
+12	MV	REAL	-10	110	0.0	1	%	Manipulated Variable	User (condition 3)	M+P_MOUT
+13										
+18	MH	REAL	-10	110	100.0	1	%	MV High Limit Value	User	M+P_MOUT
+19										
+20	ML	REAL	-10	110	0.0	1	%	MV Low Limit Value	User	M+P_MOUT
+21										

# PGS

Structure name: M+TM\_PGS

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+0	FUNC	INT	9	9	9	—	—	Tag Function Code	System	—
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG M+P_PGS
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG M+P_PGS
	b5	CAS	BOOL	FALSE	TRUE	FALSE	—	Control Mode: CASCADE	User (condition 1)	M+P_MCHG M+P_PGS
	b9	CMV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER MV	User (condition 1)	M+P_MCHG M+P_PGS
	bA	CSV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER SV	User (condition 1)	M+P_MCHG M+P_PGS
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable MANUAL	User	M+P_MCHG
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable AUTO	User	M+P_MCHG
	b5	CASI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable CASCADE	User	M+P_MCHG
	b9	CMVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER MV	User	M+P_MCHG
	bA	CSVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER SV	User	M+P_MCHG
+3	b0	MLA	BOOL	FALSE	TRUE	FALSE	—	Output Low Limit Alarm	User (condition 2)	Common
	b1	MHA	BOOL	FALSE	TRUE	FALSE	—	Output High Limit Alarm	User (condition 2)	Common
	bE	SPA	BOOL	FALSE	TRUE	FALSE	—	Stop Alarm	User (condition 2)	Common
+4	b0	MLI	BOOL	FALSE	TRUE	FALSE	—	Disable Output Low Limit Alarm	User	Common
	b1	MHI	BOOL	FALSE	TRUE	FALSE	—	Disable Output High Limit Alarm	User	Common
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	Disable All Alarms	User	Common
+5	b0	MLL	BOOL	FALSE	TRUE	FALSE	—	Alarm Level of Output Low Limit Alarm	User	—
	b1	MHL	BOOL	FALSE	TRUE	FALSE	—	Alarm Level of Output High Limit Alarm	User	—
	bE	SPL	BOOL	FALSE	TRUE	FALSE	—	Alarm Level of Stop Alarm	User	—
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—
+8	UNIT	INT	0	127	0	—	—	Unit	User	—
+9	N_	INT	0	4	1	—	—	No. of digits after the decimal point	User	—
+10	PTNO	INT	0	16	0	—	—	Number of Points	User	M+P_PGS
+12	MV	REAL	-10	110	0.0	N	%	Manipulated Variable	User (condition 3)	M+P_PGS
+13										



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+14	SV	REAL	0	999999	0.0	1	s	Setting Value	User	M+P_PGS
+15										
+16	TYP	INT	0	1	0	—	—	Operation Type	User	M+P_PGS
+18	MH	REAL	-10	110	100.0	N	%	MV High Limit Value	User	M+P_PGS
+19										
+20	ML	REAL	-10	110	0.0	N	%	MV Low Limit Value	User	M+P_PGS
+21										
+22	SV1	REAL	0	999999	0.0	1	s	Setting Time 1	User	M+P_PGS
+23										
+24	SV2	REAL	0	999999	0.0	1	s	Setting Time 2	User	M+P_PGS
+25										
+26	SV3	REAL	0	999999	0.0	1	s	Setting Time 3	User	M+P_PGS
+27										
+28	SV4	REAL	0	999999	0.0	1	s	Setting Time 4	User	M+P_PGS
+29										
+30	SV5	REAL	0	999999	0.0	1	s	Setting Time 5	User	M+P_PGS
+31										
+32	SV6	REAL	0	999999	0.0	1	s	Setting Time 6	User	M+P_PGS
+33										
+34	SV7	REAL	0	999999	0.0	1	s	Setting Time 7	User	M+P_PGS
+35										
+36	SV8	REAL	0	999999	0.0	1	s	Setting Time 8	User	M+P_PGS
+37										
+38	SV9	REAL	0	999999	0.0	1	s	Setting Time 9	User	M+P_PGS
+39										
+40	SV10	REAL	0	999999	0.0	1	s	Setting Time 10	User	M+P_PGS
+41										
+42	SV11	REAL	0	999999	0.0	1	s	Setting Time 11	User	M+P_PGS
+43										
+44	SV12	REAL	0	999999	0.0	1	s	Setting Time 12	User	M+P_PGS
+45										
+46	SV13	REAL	0	999999	0.0	1	s	Setting Time 13	User	M+P_PGS
+47										
+48	SV14	REAL	0	999999	0.0	1	s	Setting Time 14	User	M+P_PGS
+49										
+50	SV15	REAL	0	999999	0.0	1	s	Setting Time 15	User	M+P_PGS
+51										
+52	SV16	REAL	0	999999	0.0	1	s	Setting Time 16	User	M+P_PGS
+53										
+54	MV1	REAL	-10	110	0.0	N	%	Setting Output 1	User	M+P_PGS
+55										
+56	MV2	REAL	-10	110	0.0	N	%	Setting Output 2	User	M+P_PGS
+57										
+58	MV3	REAL	-10	110	0.0	N	%	Setting Output 3	User	M+P_PGS
+59										
+60	MV4	REAL	-10	110	0.0	N	%	Setting Output 4	User	M+P_PGS
+61										
+62	MV5	REAL	-10	110	0.0	N	%	Setting Output 5	User	M+P_PGS
+63										



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+64	MV6	REAL	-10	110	0.0	N	%	Setting Output 6	User	M+P_PGS
+65										
+66	MV7	REAL	-10	110	0.0	N	%	Setting Output 7	User	M+P_PGS
+67										
+68	MV8	REAL	-10	110	0.0	N	%	Setting Output 8	User	M+P_PGS
+69										
+70	MV9	REAL	-10	110	0.0	N	%	Setting Output 9	User	M+P_PGS
+71										
+72	MV10	REAL	-10	110	0.0	N	%	Setting Output 10	User	M+P_PGS
+73										
+74	MV11	REAL	-10	110	0.0	N	%	Setting Output 11	User	M+P_PGS
+75										
+76	MV12	REAL	-10	110	0.0	N	%	Setting Output 12	User	M+P_PGS
+77										
+78	MV13	REAL	-10	110	0.0	N	%	Setting Output 13	User	M+P_PGS
+79										
+80	MV14	REAL	-10	110	0.0	N	%	Setting Output 14	User	M+P_PGS
+81										
+82	MV15	REAL	-10	110	0.0	N	%	Setting Output 15	User	M+P_PGS
+83										
+84	MV16	REAL	-10	110	0.0	N	%	Setting Output 16	User	M+P_PGS
+85										

# PGS2

Structure name: M+TM\_PGS2

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+0	FUNC	INT	18	18	18	—	—	Tag Function Code	System	—
+1	b3	MAN	FALSE	TRUE	TRUE	—	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG M+P_PGS2_
	b4	AUT	FALSE	TRUE	FALSE	—	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG M+P_PGS2_
+2	b3	MANI	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable MANUAL	User	M+P_MCHG
	b4	AUTI	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable AUTO	User	M+P_MCHG
	bC	TSTPI	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable TAG STOP	User	M+P_MCHG
+3	b0	SVLA	FALSE	TRUE	FALSE	—	—	SV Low Alarm	User (condition 2)	Common
	b1	SVHA	FALSE	TRUE	FALSE	—	—	SV High Alarm	User (condition 2)	Common
	bE	SPA	FALSE	TRUE	FALSE	—	—	Stop Alarm	User (condition 2)	Common
+4	b0	SVLI	FALSE	TRUE	FALSE	—	—	Disable SV Low Alarm	User	Common
	b1	SVHI	FALSE	TRUE	FALSE	—	—	Disable SV High Alarm	User	Common
	bF	ERRI	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User	Common
+5	b0	SVLL	FALSE	TRUE	FALSE	—	—	Alarm Level of SV Low Alarm	User	—
	b1	SVHL	FALSE	TRUE	FALSE	—	—	Alarm Level of SV High Alarm	User	—
	bE	SPL	FALSE	TRUE	FALSE	—	—	Alarm Level of Stop Alarm	User	—
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—
+8	UNIT	INT	0	127	0	—	—	Unit	User	—
+9	N_	INT	0	4	1	—	—	No. of digits after the decimal point	User	—
+10	STNO	INT	0	32	0	—	—	Number of Steps Setting	User	M+P_PGS2_
+11	PVSTART	INT	0	2	0	—	—	PV Start Type	User	M+P_PGS2_
+12	SV	REAL	RL	RH	0.0	N	UNIT	Setting value	User (condition 3)	M+P_PGS2_
+13										
+14	STC	INT	0	32	0	—	—	Executing Step No.	User	M+P_PGS2_
+15	T_	INT	0	32767	0	—	s(min) <sup>2</sup>	Time in the step	User	M+P_PGS2_
+16	PV	REAL	RL	RH	0.0	N	UNIT	Process Variable	System	M+P_PGS2_
+17										
+18	SH	INT	-32768	32767	100	—	UNIT <sup>+1</sup>	SV High Limit Value	User	M+P_PGS2_
+19	SL	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	SV Low Limit Value	User	M+P_PGS2_
+20	b0	TYP_HOLD	FALSE	TRUE	TRUE	—	—	Operation Type: HOLD	User	M+P_PGS2_
	b1	TYP_RETURN	FALSE	TRUE	FALSE	—	—	Operation Type: RETURN	User	M+P_PGS2_
	b2	TYP_CYCLIC	FALSE	TRUE	FALSE	—	—	Operation Type: CYCLIC	User	M+P_PGS2_
	b8	TUNIT	BOOL	FALSE	TRUE	FALSE	—	—	Unit of time	User
+21	WAIT	INT	0	32767	0	—	UNIT <sup>+1</sup>	Wait Width	User	M+P_PGS2_

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+22	RH	INT	-32768	32767	100	—	UNIT <sup>+1</sup>	Engineering Value High Limit	User	M+P_PGS2_
+23	RL	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Engineering Value Low Limit	User	M+P_PGS2_
+26	SV0	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Start Point	User	M+P_PGS2_
+27	SV0C	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Start Point (Current)	System	M+P_PGS2_
+28	T1_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 1	User	M+P_PGS2_
+29	SV1	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 1	User	M+P_PGS2_
+30	T2_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 2	User	M+P_PGS2_
+31	SV2	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 2	User	M+P_PGS2_
+32	T3_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 3	User	M+P_PGS2_
+33	SV3	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 3	User	M+P_PGS2_
+34	T4_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 4	User	M+P_PGS2_
+35	SV4	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 4	User	M+P_PGS2_
+36	T5_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 5	User	M+P_PGS2_
+37	SV5	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 5	User	M+P_PGS2_
+38	T6_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 6	User	M+P_PGS2_
+39	SV6	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 6	User	M+P_PGS2_
+40	T7_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 7	User	M+P_PGS2_
+41	SV7	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 7	User	M+P_PGS2_
+42	T8_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 8	User	M+P_PGS2_
+43	SV8	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 8	User	M+P_PGS2_
+44	T9_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 9	User	M+P_PGS2_
+45	SV9	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 9	User	M+P_PGS2_
+46	T10_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 10	User	M+P_PGS2_
+47	SV10	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 10	User	M+P_PGS2_
+48	T11_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 11	User	M+P_PGS2_
+49	SV11	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 11	User	M+P_PGS2_
+50	T12_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 12	User	M+P_PGS2_
+51	SV12	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 12	User	M+P_PGS2_
+52	T13_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 13	User	M+P_PGS2_
+53	SV13	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 13	User	M+P_PGS2_
+54	T14_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 14	User	M+P_PGS2_
+55	SV14	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 14	User	M+P_PGS2_
+56	T15_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 15	User	M+P_PGS2_
+57	SV15	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 15	User	M+P_PGS2_
+58	T16_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 16	User	M+P_PGS2_
+59	SV16	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 16	User	M+P_PGS2_
+60	T17_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 17	User	M+P_PGS2_
+61	SV17	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 17	User	M+P_PGS2_
+62	T18_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 18	User	M+P_PGS2_
+63	SV18	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 18	User	M+P_PGS2_
+64	T19_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 19	User	M+P_PGS2_
+65	SV19	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 19	User	M+P_PGS2_
+66	T20_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 20	User	M+P_PGS2_
+67	SV20	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 20	User	M+P_PGS2_
+68	T21_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 21	User	M+P_PGS2_
+69	SV21	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 21	User	M+P_PGS2_
+70	T22_	INT	0	32767	0	—	s(min) <sup>+2</sup>	Time Span of Step 22	User	M+P_PGS2_
+71	SV22	INT	-32768	32767	0	—	UNIT <sup>+1</sup>	Setting Value of Step 22	User	M+P_PGS2_



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+72	T23_	INT	0	32767	0	—	s(min) <sup>*2</sup>	Time Span of Step 23	User	M+P_PGS2_	
+73	SV23	INT	-32768	32767	0	—	UNIT <sup>*1</sup>	Setting Value of Step 23	User	M+P_PGS2_	
+74	T24_	INT	0	32767	0	—	s(min) <sup>*2</sup>	Time Span of Step 24	User	M+P_PGS2_	
+75	SV24	INT	-32768	32767	0	—	UNIT <sup>*1</sup>	Setting Value of Step 24	User	M+P_PGS2_	
+76	T25_	INT	0	32767	0	—	s(min) <sup>*2</sup>	Time Span of Step 25	User	M+P_PGS2_	
+77	SV25	INT	-32768	32767	0	—	UNIT <sup>*1</sup>	Setting Value of Step 25	User	M+P_PGS2_	
+78	T26_	INT	0	32767	0	—	s(min) <sup>*2</sup>	Time Span of Step 26	User	M+P_PGS2_	
+79	SV26	INT	-32768	32767	0	—	UNIT <sup>*1</sup>	Setting Value of Step 26	User	M+P_PGS2_	
+80	T27_	INT	0	32767	0	—	s(min) <sup>*2</sup>	Time Span of Step 27	User	M+P_PGS2_	
+81	SV27	INT	-32768	32767	0	—	UNIT <sup>*1</sup>	Setting Value of Step 27	User	M+P_PGS2_	
+82	T28_	INT	0	32767	0	—	s(min) <sup>*2</sup>	Time Span of Step 28	User	M+P_PGS2_	
+83	SV28	INT	-32768	32767	0	—	UNIT <sup>*1</sup>	Setting Value of Step 28	User	M+P_PGS2_	
+84	T29_	INT	0	32767	0	—	s(min) <sup>*2</sup>	Time Span of Step 29	User	M+P_PGS2_	
+85	SV29	INT	-32768	32767	0	—	UNIT <sup>*1</sup>	Setting Value of Step 29	User	M+P_PGS2_	
+86	T30_	INT	0	32767	0	—	s(min) <sup>*2</sup>	Time Span of Step 30	User	M+P_PGS2_	
+87	SV30	INT	-32768	32767	0	—	UNIT <sup>*1</sup>	Setting Value of Step 30	User	M+P_PGS2_	
+88	T31_	INT	0	32767	0	—	s(min) <sup>*2</sup>	Time Span of Step 31	User	M+P_PGS2_	
+89	SV31	INT	-32768	32767	0	—	UNIT <sup>*1</sup>	Setting Value of Step 31	User	M+P_PGS2_	
+90	T32_	INT	0	32767	0	—	s(min) <sup>*2</sup>	Time Span of Step 32	User	M+P_PGS2_	
+91	SV32	INT	-32768	32767	0	—	UNIT <sup>*1</sup>	Setting Value of Step 32	User	M+P_PGS2_	
+94	b0	DOM_ADV_START	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Advance Command	Tag data access control	—
	bD	TSTP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: TAG STOP	Tag data access control	—
+95	b0	DIM_WAIT_MODE	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Waiting	System	—

\*1 This tag data is set in integer, ignoring the number of digits after the decimal point (N\_).

When the setting after the decimal point is required depending on the engineering value range, set the tag data as follows.

(Example) When the setting value is 1.5MPa, convert its unit to 1500KPa to fit the value within the range of -32768 to 32767.

\*2 When the unit of time (TUNIT) is FALSE, "s" is used and "min" is used for TRUE.

# PVAL

Structure name: M+TM\_PVAL

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+0	FUNC	INT	23	23	23	—	—	Tag Function Code	System	—
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG
	b5	CAS	BOOL	FALSE	TRUE	FALSE	—	Control Mode: CASCADE	User (condition 1)	M+P_MCHG
	bA	CSV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER SV	User (condition 1)	M+P_MCHG
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable MANUAL	User	—
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable AUTO	User	—
	b5	CASI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable CASCADE	User	—
	bA	CSVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER SV	User	—
	bC	TSTPI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable TAG STOP	User	—
	bE	OVRI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable OVERRIDE	User	—
	bF	SIMI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable SIMULATION	User	—
+3	b0	TOA	BOOL	FALSE	TRUE	FALSE	—	Time-out Alarm	User (condition 2)	Common
	b1	TRIPA	BOOL	FALSE	TRUE	FALSE	—	Trip Alarm	User (condition 2)	Common
	b2	DVLA	BOOL	FALSE	TRUE	FALSE	—	Large Deviation Alarm	User (condition 2)	Common
	b3	DPNA	BOOL	FALSE	TRUE	FALSE	—	Negative Variation Rate Alarm	User (condition 2)	Common
	b4	DPPA	BOOL	FALSE	TRUE	FALSE	—	Positive Variation Rate Alarm	User (condition 2)	Common
	b5	PLA	BOOL	FALSE	TRUE	FALSE	—	Input Low Limit Alarm	User (condition 2)	Common
	b6	PHA	BOOL	FALSE	TRUE	FALSE	—	Input High Limit Alarm	User (condition 2)	Common
	b7	LLA	BOOL	FALSE	TRUE	FALSE	—	Input Low Low Limit Alarm	User (condition 2)	Common
	b8	HHA	BOOL	FALSE	TRUE	FALSE	—	Input High High Limit Alarm	User (condition 2)	Common
	b9	SEA	BOOL	FALSE	TRUE	FALSE	—	Sensor Error Alarm	User (condition 2)	Common
bE	SPA	BOOL	FALSE	TRUE	FALSE	—	Stop Alarm	User (condition 2)	Common	



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+4	b0	TOI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Time-out Alarm	User	Common
	b1	TRIP1	BOOL	FALSE	TRUE	FALSE	—	—	Disable Trip Alarm	User	Common
	b2	DVLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Large Deviation Alarm	User	Common
	b3	DPNI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Negative Variation Rate Alarm	User	Common
	b4	DPPI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Positive Variation Rate Alarm	User	Common
	b5	PLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Limit Alarm	User	Common
	b6	PHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High Limit Alarm	User	Common
	b7	LLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input Low Low Limit Alarm	User	Common
	b8	HHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Input High High Limit Alarm	User	Common
	b9	SEI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Sensor Error Alarm	User	Common
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User	Common
+5	b0	TOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Time-out Alarm	User	Common
	b1	TRIP1	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Trip Alarm	User	Common
	b2	DVLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Large Deviation Alarm	User	Common
	b3	DPNL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Negative Variation Rate Alarm	User	Common
	b4	DPPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Positive Variation Rate Alarm	User	Common
	b5	PLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Limit Alarm	User	Common
	b6	PHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High Limit Alarm	User	Common
	b7	LLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input Low Low Limit Alarm	User	Common
	b8	HHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Input High High Limit Alarm	User	Common
	b9	SENL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Sensor Error Alarm	User	Common
	bE	SPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Stop Alarm	User	Common
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—	
+8	FPNO	INT	1	50	1	—	—	Faceplate Display Pattern	User	—	
+10	PV	REAL	0	100	0.0	1	%	Motor Valve Opening	System	—	
+11											
+12	b0	VOUT_OPEN	BOOL	FALSE	TRUE	FALSE	—	—	Output of Open Command Signal	System	—
	b1	VOUT_CLOSE	BOOL	FALSE	TRUE	FALSE	—	—	Output of Close Command Signal	System	—
+14	SVC	REAL	0	100	0.0	1	%	Setting Value of Valve Opening (Current)	System	—	
+15											

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+16	DV	REAL	-100	100	0.0	1	%	Deviation of Valve Opening	System	—	
+17											
+18	HS0	REAL	0	100	0.0	1	%	Hysteresis	User	—	
+19											
+20	DBND	REAL	0	100	0.0	1	%	Dead Band	User	—	
+21											
+26	PH_	REAL	0	100	100.0	1	%	PV High Limit Alarm Value	User	—	
+27											
+28	PL	REAL	0	100	0.0	1	%	PV Low Limit Alarm Value	User	—	
+29											
+30	HH	REAL	0	100	100.0	1	%	PV High High Limit Alarm Value	User	—	
+31											
+32	LL	REAL	0	100	0.0	1	%	PV Low Low Limit Alarm Value	User	—	
+33											
+34	SH	REAL	0	100	100.0	1	%	SV High Limit Value	User	—	
+35											
+36	SL	REAL	0	100	0.0	1	%	SV Low Limit Value	User	—	
+37											
+38	ALPHA	REAL	0	1	0.2	2	—	PV Filter Coefficient	User	M+P_IN	
+39											
+40	HS	REAL	0	100	0.0	1	%	PV High/Low Limit Alarm Hysteresis	User	—	
+41											
+42	CTIM	REAL	0	9999	0.0	2	s	Variation Rate Alarm Check Time	User	—	
+43											
+44	DPL	REAL	0	100	100.0	1	%	Variation Rate Alarm Value	User	—	
+45											
+50	DVL	REAL	0	100	100.0	1	%	Deviation Limit Value	User	—	
+51											
+83	TOT	INT	0	99	5	—	s	Time-out Timer	User	—	
+84	DOT	REAL	0	9.0	1.0	1	s	Command Pulse Period	User	—	
+85											
+86	SIMT	INT	0	99	3	—	s	Simulation Answer Period	User	—	
+87	bD	SVLA	BOOL	FALSE	TRUE	FALSE	—	—	SV Low Alarm	User (condition 2)	Common
	bE	SVHA	BOOL	FALSE	TRUE	FALSE	—	—	SV High Alarm	User (condition 2)	Common
	bF	DSVLA	BOOL	FALSE	TRUE	FALSE	—	—	SV Variation Rate Limit Alarm	User (condition 2)	Common
+88	bD	SVLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable SV Low Alarm	User	Common
	bE	SVHI	BOOL	FALSE	TRUE	FALSE	—	—	Disable SV High Alarm	User	Common
	bF	DSVLI	BOOL	FALSE	TRUE	FALSE	—	—	Disable SV Variation Rate Limit Alarm	User	Common
+89	bD	SVLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of SV Low Alarm	User	Common
	bE	SVHL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of SV High Alarm	User	Common
	bF	DSVLL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of SV Variation Rate Limit Alarm	User	Common



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+90	SV	REAL	0	100	0.0	1	%	Setting Value of Valve Opening (Target)	User	—	
+91											
+92	DSVL	REAL	0	100	100.0	1	%	SV Variation Rate High Limit Value	User	—	
+93											
+94	b0	DOM_OPEN_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Open by PC	Tag data access control	—
	b1	DOM_CLOSE_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Close by PC	Tag data access control	—
	b2	DOM_STOP_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Stop by PC	Tag data access control	—
	b5	DOM_TO_RESET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Time-out Reset	Tag data access control	—
	bD	TSTP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: TAG STOP	Tag data access control	—
	bE	OVR	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: OVERRIDE	Tag data access control	—
	bF	SIM	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: SIMULATION	Tag data access control	—
+95	b0	DIM_OPEN	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Status Answer	System	—
	b1	DIM_CLOSE	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Status Answer	System	—
	b3	DIM_LOCAL	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Local	System	—
	b4	DIM_REMOTE	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Remote	System	—



# HTCL

Structure name: M+TM\_HTCL

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB
			Low limit	High limit						
+0	FUNC	INT	24	24	24	—	—	Tag Function Code	System	—
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	Control Mode: MANUAL	User (condition 1)	M+P_MCHG
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	Control Mode: AUTO	User (condition 1)	M+P_MCHG
	b5	CAS	BOOL	FALSE	TRUE	FALSE	—	Control Mode: CASCADE	User (condition 1)	M+P_MCHG
	b9	CMV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER MV	User (condition 1)	M+P_MCHG
	bA	CSV	BOOL	FALSE	TRUE	FALSE	—	Control Mode: COMPUTER SV	User (condition 1)	M+P_MCHG
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable MANUAL	User	—
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable AUTO	User	—
	b5	CASI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable CASCADE	User	—
	b9	CMVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER MV	User	—
	bA	CSVI	BOOL	FALSE	TRUE	TRUE	—	Disable Mode Change: Disable COMPUTER SV	User	—
	bC	TSTPI	BOOL	FALSE	TRUE	FALSE	—	Disable Mode Change: Disable TAG STOP	User	—
+3	b0	MLA_HT	BOOL	FALSE	TRUE	FALSE	—	Heating Output Low Limit Alarm	User (condition 2)	Common
	b1	MHA_HT	BOOL	FALSE	TRUE	FALSE	—	Heating Output High Limit Alarm	User (condition 2)	Common
	b2	DMLA_HT	BOOL	FALSE	TRUE	FALSE	—	Heating Output Variation Rate Limit Alarm	User (condition 2)	Common
	b3	MLA_CL	BOOL	FALSE	TRUE	FALSE	—	Cooling Output Low Limit Alarm	User (condition 2)	Common
	b4	MHA_CL	BOOL	FALSE	TRUE	FALSE	—	Cooling Output High Limit Alarm	User (condition 2)	Common
	b5	DMLA_CL	BOOL	FALSE	TRUE	FALSE	—	Cooling Output Variation Rate Limit Alarm	User (condition 2)	Common
	bA	OOA	BOOL	FALSE	TRUE	FALSE	—	Output Open Alarm	User (condition 2)	Common
	bB	HBOA	BOOL	FALSE	TRUE	FALSE	—	Heater Burnout Alarm	User (condition 2)	Common
	bE	SPA	BOOL	FALSE	TRUE	FALSE	—	Stop Alarm	User (condition 2)	Common



Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+4	b0	MLI_HT	BOOL	FALSE	TRUE	FALSE	—	—	Disable Heating Output Low Limit Alarm	User	Common
	b1	MHI_HT	BOOL	FALSE	TRUE	FALSE	—	—	Disable Heating Output High Limit Alarm	User	Common
	b2	DMLI_HT	BOOL	FALSE	TRUE	FALSE	—	—	Disable Heating Output Variation Rate Limit Alarm	User	Common
	b3	MLI_CL	BOOL	FALSE	TRUE	FALSE	—	—	Disable Cooling Output Low Limit Alarm	User	Common
	b4	MHI_CL	BOOL	FALSE	TRUE	FALSE	—	—	Disable Cooling Output High Limit Alarm	User	Common
	b5	DMLI_CL	BOOL	FALSE	TRUE	FALSE	—	—	Disable Cooling Output Variation Rate Limit Alarm	User	Common
	bB	HBOI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Heater Burnout Alarm	User	Common
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User	Common
+5	b0	MLL_HT	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Heating Output Low Limit Alarm	User	—
	b1	MHL_HT	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Heating Output High Limit Alarm	User	—
	b2	DMLL_HT	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Heating Output Variation Rate Limit Alarm	User	—
	b3	MLL_CL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Cooling Output Low Limit Alarm	User	—
	b4	MHL_CL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Cooling Output High Limit Alarm	User	—
	b5	DMLL_CL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Cooling Output Variation Rate Limit Alarm	User	—
	bA	OOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Output Open Alarm	User	—
	bB	HBOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Heater Burnout Alarm	User	—
	bE	SPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Stop Alarm	User	—
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	—	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	—	
+8	FPNO	INT	1	50	1	—	—	Faceplate Display Pattern	User	—	
+9	PRM_TRK	INT	0	1	0	—	—	Tracking of PID Parameters	User	—	
+10	MV_HT	REAL	-10	110	0.0	1	%	Heating Manipulated Variable	User	—	
+11											
+12	MV_CL	REAL	-10	110	0.0	1	%	Cooling Manipulated Variable	User	—	
+13											
+14	SV	REAL	0	100	0.0	1	%	Setting value	User	—	
+15											

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	Tag access FB	
			Low limit	High limit							
+26	MH_HT	REAL	-10	110	100.0	1	%	Heating MV High Limit Value	User	—	
+27											
+28	ML_HT	REAL	-10	110	0.0	1	%	Heating MV Low Limit Value	User	—	
+29											
+30	MH_CL	REAL	-10	110	100.0	1	%	Cooling MV High Limit Value	User	—	
+31											
+32	ML_CL	REAL	-10	110	0.0	1	%	Cooling MV Low Limit Value	User	—	
+33											
+34	SH	REAL	0	100	100.0	1	%	SV High Limit Value	User	—	
+35											
+36	SL	REAL	0	100	0.0	1	%	SV Low Limit Value	User	—	
+37											
+48	DML_HT	REAL	0	100	100.0	1	%	Heating Output Variation Rate High Limit Value	User	—	
+49											
+50	DML_CL	REAL	0	100	100.0	1	%	Cooling Output Variation Rate High Limit Value	User	—	
+51											
+52	P_HT	REAL	0	999	1.0	2	—	Heating Gain	User	—	
+53											
+54	I_HT	REAL	0	9999	10.0	1	s	Heating Integral Time	User	—	
+55											
+56	D_HT	REAL	0	9999	0.0	1	s	Heating Derivative Time	User	—	
+57											
+58	P_CL	REAL	0	999	1.0	2	—	Cooling Gain	User	—	
+59											
+60	I_CL	REAL	0	9999	10.0	1	s	Cooling Integral Time	User	—	
+61											
+62	D_CL	REAL	0	9999	0.0	1	s	Cooling Derivative Time	User	—	
+63											
+64	DBND	REAL	-100	100	0.0	1	%	Dead Band	User	—	
+65											
+66	HS	REAL	0	50	0.0	1	%	Hysteresis	User	—	
+67											
+68	SPLT	REAL	0	100	50.0	1	%	Split Value	User	—	
+69											
+86	PRM_SEL	INT	0	3	0	—	—	Target to Reflect Results of Auto Tuning	User	—	
+94	bD	TSTP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: TAG STOP	Tag data access control	—
+95	b9	DIM_PID_HT	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Heating PID Parameters	System	—
	bA	DIM_PID_CL	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Cooling PID Parameters	System	—

# NREV

Structure name: M+TM\_NREV

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	
			Low limit	High limit						
+0	FUNC	INT	128	128	128	—	—	Tag Function Code	System	
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	—	Control Mode: MANUAL	User (condition 1)
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: AUTO	User (condition 1)
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable MANUAL	User
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable AUTO	User
	bE	OVRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable OVERRIDE	User
	bF	SIMI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable SIMULATION	User
+3	b0	TOA	BOOL	FALSE	TRUE	FALSE	—	—	Time-out Alarm	User (condition 2)
	b1	TRIPA	BOOL	FALSE	TRUE	FALSE	—	—	Trip Alarm	User (condition 2)
+4	b0	TOI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Time-out Alarm	User
	b1	TRIP1	BOOL	FALSE	TRUE	FALSE	—	—	Disable Trip Alarm	User
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User
+5	b0	TOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Time-out Alarm	User
	b1	TRIPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Trip Alarm	User
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	
+8	FPNO	INT	1	50	1	—	—	Faceplate Display Pattern	User	
+9	b0	DOM_RUN_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Operation by PC	Tag data access control
	b1	DOM_STOP_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Stop by PC	Tag data access control
	b5	DOM_TO_RESET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Time-out Reset	Tag data access control
	bE	OVR	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: OVERRIDE	Tag data access control
	bF	SIM	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: SIMULATION	Tag data access control
+10	b0	DIM_RUN	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Status Answer	System
	b1	DIM_STOP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Status Answer	System
	b3	DIM_LOCAL	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Local	System
	b4	DIM_REMOTE	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Remote	System
+14	TOT	INT	0	99	5	—	s	Time-out Timer	User	
+15	DOT	INT	0	9	1	—	s	Command Pulse Period	User	
+16	SIMT	INT	0	99	3	—	s	Simulation Answer Period	User	

# REV

Structure name: M+TM\_REV

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	
			Low limit	High limit						
+0	FUNC	INT	129	129	129	—	—	Tag Function Code	System	
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	—	Control Mode: MANUAL	User (condition 1)
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: AUTO	User (condition 1)
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable MANUAL	User
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable AUTO	User
	bE	OVRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable OVERRIDE	User
	bF	SIMI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable SIMULATION	User
+3	b0	TOA	BOOL	FALSE	TRUE	FALSE	—	—	Time-out Alarm	User (condition 2)
	b1	TRIPA	BOOL	FALSE	TRUE	FALSE	—	—	Trip Alarm	User (condition 2)
+4	b0	TOI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Time-out Alarm	User
	b1	TRIP1	BOOL	FALSE	TRUE	FALSE	—	—	Disable Trip Alarm	User
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User
+5	b0	TOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Time-out Alarm	User
	b1	TRIPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Trip Alarm	User
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	
+8	FPNO	INT	1	50	1	—	—	Faceplate Display Pattern	User	
+9	b0	DOM_FWD_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Forward Run by PC	Tag data access control
	b1	DOM_STOP_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Stop by PC	Tag data access control
	b2	DOM_REV_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Reverse Run by PC	Tag data access control
	b5	DOM_TO_RESET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Time-out Reset	Tag data access control
	bE	OVR	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: OVERRIDE	Tag data access control
	bF	SIM	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: SIMULATION	Tag data access control
+10	b0	DIM_FWD	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Status Answer	System
	b1	DIM_STOP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Status Answer	System
	b2	DIM_REV	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Status Answer	System
	b3	DIM_LOCAL	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Local	System
	b4	DIM_REMOTE	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Remote	System
+14	TOT	INT	0	99	5	—	s	Time-out Timer	User	
+15	DOT	INT	0	9	1	—	s	Command Pulse Period	User	
+16	SIMT	INT	0	99	3	—	s	Simulation Answer Period	User	



# MVAL1

Structure name: M+TM\_MVAL1

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	
			Low limit	High limit						
+0	FUNC	INT	130	130	130	—	—	Tag Function Code	System	
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	—	Control Mode: MANUAL	User (condition 1)
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: AUTO	User (condition 1)
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable MANUAL	User
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable AUTO	User
	bE	OVRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable OVERRIDE	User
	bF	SIMI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable SIMULATION	User
+3	b0	TOA	BOOL	FALSE	TRUE	FALSE	—	—	Time-out Alarm	User (condition 2)
	b1	TRIPA	BOOL	FALSE	TRUE	FALSE	—	—	Trip Alarm	User (condition 2)
+4	b0	TOI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Time-out Alarm	User
	b1	TRIP1	BOOL	FALSE	TRUE	FALSE	—	—	Disable Trip Alarm	User
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User
+5	b0	TOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Time-out Alarm	User
	b1	TRIPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Trip Alarm	User
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	
+8	FPNO	INT	1	50	1	—	—	Faceplate Display Pattern	User	
+9	b0	DOM_OPEN_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Open by PC	Tag data access control
	b2	DOM_CLOSE_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Close by PC	Tag data access control
	b5	DOM_TO_RESET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Time-out Reset	Tag data access control
	bE	OVR	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: OVERRIDE	Tag data access control
	bF	SIM	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: SIMULATION	Tag data access control
+10	b0	DIM_OPEN	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Status Answer	System
	b1	DIM_SEMI_CLOSE	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Status Answer	System
	b2	DIM_CLOSE	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Status Answer	System
	b3	DIM_LOCAL	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Local	System
	b4	DIM_REMOTE	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Remote	System
+14	TOT	INT	0	99	5	—	s	Time-out Timer	User	
+15	DOT	INT	0	9	1	—	s	Command Pulse Period	User	
+16	SIMT	INT	0	99	3	—	s	Simulation Answer Period	User	

# MVAL2

Structure name: M+TM\_MVAL2

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	
			Low limit	High limit						
+0	FUNC	INT	131	131	131	—	—	Tag Function Code	System	
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	—	Control Mode: MANUAL	User (condition 1)
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: AUTO	User (condition 1)
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable MANUAL	User
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable AUTO	User
	bE	OVRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable OVERRIDE	User
	bF	SIMI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable SIMULATION	User
+3	b0	TOA	BOOL	FALSE	TRUE	FALSE	—	—	Time-out Alarm	User (condition 2)
	b1	TRIPA	BOOL	FALSE	TRUE	FALSE	—	—	Trip Alarm	User (condition 2)
+4	b0	TOI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Time-out Alarm	User
	b1	TRIP1	BOOL	FALSE	TRUE	FALSE	—	—	Disable Trip Alarm	User
	bF	ERRI	BOOL	FALSE	TRUE	FALSE	—	—	Disable All Alarms	User
+5	b0	TOL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Time-out Alarm	User
	b1	TRIPL	BOOL	FALSE	TRUE	FALSE	—	—	Alarm Level of Trip Alarm	User
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	
+8	FPNO	INT	1	50	1	—	—	Faceplate Display Pattern	User	
+9	b0	DOM_OPEN_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Open by PC	Tag data access control
	b1	DOM_STOP_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Stop by PC	Tag data access control
	b2	DOM_CLOSE_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Close by PC	Tag data access control
	b5	DOM_TO_RESET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Time-out Reset	Tag data access control
	bE	OVR	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: OVERRIDE	Tag data access control
	bF	SIM	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: SIMULATION	Tag data access control
+10	b0	DIM_OPEN	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Status Answer	System
	b1	DIM_SEMI_CLOSE	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Status Answer	System
	b2	DIM_CLOSE	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Status Answer	System
	b3	DIM_LOCAL	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Local	System
	b4	DIM_REMOTE	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Remote	System
+14	TOT	INT	0	99	5	—	s	Time-out Timer	User	
+15	DOT	INT	0	9	1	—	s	Command Pulse Period	User	
+16	SIMT	INT	0	99	3	—	s	Simulation Answer Period	User	

# TIMER1

Structure name: M+TM\_TIMER1

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	
			Low limit	High limit						
+0	FUNC	INT	132	132	132	—	—	Tag Function Code	System	
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	
+9	b0	DOM_RUN_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Start by PC	Tag data access control
	b1	DOM_STOP_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Stop by PC	Tag data access control
	b2	DOM_RESET_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Reset by PC	Tag data access control
	b3	DOM_RESET_START_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Reset/Start by PC	Tag data access control
+10	b0	DIM_RUN	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Start	System
	b1	DIM_STOP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Stop	System
	b2	DIM_RESET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Reset	System
	b6	DIM_PRE_COMP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Pre-complete External Output	System
	b7	DIM_COMP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Complete External Output	System
+14	UNIT	INT	0	127	0	—	—	Unit	User	
+16	PV	DINT	RL	RH	0	—	UNIT	Process Variable	System	
+17										
+18	PSV	DINT	RL	RH	0	—	UNIT	Setting Value (Preset)	User	
+19										
+20	SV	DINT	RL	RH	0	—	UNIT	Setting value	User	
+21										
+22	MULT	INT	0	1	1	—	—	Multiplying Factor (0: second, 1: minute)	User	
+24	RH	DINT	0	99999999	99999999	—	UNIT	Timer High Limit	User	
+25										
+26	RL	DINT	0	99999999	0	—	UNIT	Timer Low Limit	User	
+27										



# TIMER2

Structure name: M+TM\_TIMER2

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	
			Low limit	High limit						
+0	FUNC	INT	133	133	133	—	—	Tag Function Code	System	
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	
+9	b0	DOM_RUN_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Start by PC	Tag data access control
	b1	DOM_STOP_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Stop by PC	Tag data access control
	b2	DOM_RESET_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Reset by PC	Tag data access control
	b3	DOM_RESET_START_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Reset/Start by PC	Tag data access control
+10	b0	DIM_RUN	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Start	System
	b1	DIM_STOP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Stop	System
	b2	DIM_RESET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Reset	System
	b6	DIM_PRE_COMP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Pre-complete External Output	System
	b7	DIM_COMP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Complete External Output	System
+14	UNIT	INT	0	127	0	—	—	Unit	User	
+16	PV	DINT	RL	RH	0	—	UNIT	Process Variable	System	
+17										
+18	PSV	DINT	RL	RH	0	—	UNIT	Setting Value (Preset)	User	
+19										
+20	SV	DINT	RL	RH	0	—	UNIT	Setting value	User	
+21										
+22	MULT	INT	0	1	1	—	—	Multiplying Factor (0: second, 1: minute)	User	
+24	RH	DINT	0	99999999	99999999	—	UNIT	Timer High Limit	User	
+25										
+26	RL	DINT	0	99999999	0	—	UNIT	Timer Low Limit	User	
+27										



# COUNT1

Structure name: M+TM\_COUNT1

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	
			Low limit	High limit						
+0	FUNC	INT	134	134	134	—	—	Tag Function Code	System	
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	
+9	b0	DOM_RUN_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Start by PC	Tag data access control
	b1	DOM_STOP_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Stop by PC	Tag data access control
	b2	DOM_RESET_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Reset by PC	Tag data access control
	b3	DOM_RESET_START_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Reset/Start by PC	Tag data access control
+10	b0	DIM_RUN	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Start	System
	b1	DIM_STOP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Stop	System
	b2	DIM_RESET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Reset	System
	b6	DIM_PRE_COMP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Pre-complete External Output	System
	b7	DIM_COMP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Complete External Output	System
+14	UNIT	INT	0	127	0	—	—	Unit	User	
+16	PV	DINT	RL	RH	0	—	UNIT	Process Variable	System	
+17										
+18	PSV	DINT	RL	RH	0	—	UNIT	Setting Value (Preset)	User	
+19										
+20	SV	DINT	RL	RH	0	—	UNIT	Setting value	User	
+21										
+22	MULT	INT	1	999	1	—	—	Multiplying Factor	User	
+24	RH	DINT	0	99999999	99999999	—	UNIT	Counter High Limit	User	
+25										
+26	RL	DINT	0	99999999	0	—	UNIT	Counter Low Limit	User	
+27										

# COUNT2

Structure name: M+TM\_COUNT2

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	
			Low limit	High limit						
+0	FUNC	INT	135	135	135	—	—	Tag Function Code	System	
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	
+9	b0	DOM_RUN_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Start by PC	Tag data access control
	b1	DOM_STOP_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Stop by PC	Tag data access control
	b2	DOM_RESET_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Reset by PC	Tag data access control
	b3	DOM_RESET_START_SET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Reset/Start by PC	Tag data access control
+10	b0	DIM_RUN	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Start	System
	b1	DIM_STOP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Stop	System
	b2	DIM_RESET	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Reset	System
	b6	DIM_PRE_COMP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Pre-complete External Output	System
	b7	DIM_COMP	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Complete External Output	System
+14	UNIT	INT	0	127	0	—	—	Unit	User	
+16	PV	DINT	RL	RH	0	—	UNIT	Process Variable	System	
+17										
+18	PSV	DINT	RL	RH	0	—	UNIT	Setting Value (Preset)	User	
+19										
+20	SV	DINT	RL	RH	0	—	UNIT	Setting value	User	
+21										
+22	MULT	INT	1	999	1	—	—	Multiplying Factor	User	
+24	RH	DINT	0	99999999	99999999	—	UNIT	Counter High Limit	User	
+25										
+26	RL	DINT	0	99999999	0	—	UNIT	Counter Low Limit	User	
+27										



# PB

Structure name: M+TM\_PB

Offset	Label	Data type	Setting/Storage range		Initial value	Number of digits after the decimal point	Unit	Application	Storage	
			Low limit	High limit						
+0	FUNC	INT	136	136	136	—	—	Tag Function Code	System	
+1	b3	MAN	BOOL	FALSE	TRUE	TRUE	—	—	Control Mode: MANUAL	User (condition 1)
	b4	AUT	BOOL	FALSE	TRUE	FALSE	—	—	Control Mode: AUTO	User (condition 1)
+2	b3	MANI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable MANUAL	User
	b4	AUTI	BOOL	FALSE	TRUE	FALSE	—	—	Disable Mode Change: Disable AUTO	User
+6	CTNO	INT	0	32	0	—	—	Lockout Tag No.	System	
+7	CTFN	WORD	0	H0002	H0000	—	—	Lockout Tag Function	System	
+9	b0	DOM_SET1	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Command 1 by PC	Tag data access control
	b1	DOM_SET2	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Command 2 by PC	Tag data access control
	b2	DOM_SET3	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Command 3 by PC	Tag data access control
	b3	DOM_SET4	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Command 4 by PC	Tag data access control
	b4	DOM_SET5	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Output Buffer: Command 5 by PC	Tag data access control
+10	b0	DIM_ON1	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Status 1 Answer	System
	b1	DIM_ON2	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Status 2 Answer	System
	b2	DIM_ON3	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Status 3 Answer	System
	b3	DIM_ON4	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Status 4 Answer	System
	b4	DIM_ON5	BOOL	FALSE	TRUE	FALSE	—	—	Monitor Input Buffer: Status 5 Answer	System
+15	DOT	INT	0	9	1	—	s	Command Pulse Period	User	
+17	b0	FPINH1	BOOL	FALSE	TRUE	FALSE	—	—	Disable Display: Faceplate Button 1	User
	b1	FPINH2	BOOL	FALSE	TRUE	FALSE	—	—	Disable Display: Faceplate Button 2	User
	b2	FPINH3	BOOL	FALSE	TRUE	FALSE	—	—	Disable Display: Faceplate Button 3	User
	b3	FPINH4	BOOL	FALSE	TRUE	FALSE	—	—	Disable Display: Faceplate Button 4	User
	b4	FPINH5	BOOL	FALSE	TRUE	FALSE	—	—	Disable Display: Faceplate Button 5	User
+18	b0	BTNINH1	BOOL	FALSE	TRUE	FALSE	—	—	Disable Control Button: Button 1	User
	b1	BTNINH2	BOOL	FALSE	TRUE	FALSE	—	—	Disable Control Button: Button 2	User
	b2	BTNINH3	BOOL	FALSE	TRUE	FALSE	—	—	Disable Control Button: Button 3	User
	b3	BTNINH4	BOOL	FALSE	TRUE	FALSE	—	—	Disable Control Button: Button 4	User
	b4	BTNINH5	BOOL	FALSE	TRUE	FALSE	—	—	Disable Control Button: Button 5	User
+19	FPNO1	INT	1	10000	1	—	—	Faceplate Display 1 Pattern	User	
+20	FPNO2	INT	1	10000	1	—	—	Faceplate Display 2 Pattern	User	
+21	FPNO3	INT	1	10000	1	—	—	Faceplate Display 3 Pattern	User	
+22	FPNO4	INT	1	10000	1	—	—	Faceplate Display 4 Pattern	User	
+23	FPNO5	INT	1	10000	1	—	—	Faceplate Display 5 Pattern	User	

# ALM

Structure name: M+TM\_ALM

Offset	Label	Data type	Setting/Storage range		Initial value	Unit	Application	Storage	
			Low limit	High limit					
+0	FUNC	INT	256	256	256	—	Tag Function Code	System	
+1	b0	ALM1	BOOL	FALSE	TRUE	FALSE	—	Alarm 1	System
	b1	ALM2	BOOL	FALSE	TRUE	FALSE	—	Alarm 2	System
	b2	ALM3	BOOL	FALSE	TRUE	FALSE	—	Alarm 3	System
	b3	ALM4	BOOL	FALSE	TRUE	FALSE	—	Alarm 4	System
	b4	ALM5	BOOL	FALSE	TRUE	FALSE	—	Alarm 5	System
	b5	ALM6	BOOL	FALSE	TRUE	FALSE	—	Alarm 6	System
	b6	ALM7	BOOL	FALSE	TRUE	FALSE	—	Alarm 7	System
	b7	ALM8	BOOL	FALSE	TRUE	FALSE	—	Alarm 8	System
+2	b0	ALML1	BOOL	FALSE	TRUE	FALSE	—	Alarm Level of Alarm 1	User
	b1	ALML2	BOOL	FALSE	TRUE	FALSE	—	Alarm Level of Alarm 2	User
	b2	ALML3	BOOL	FALSE	TRUE	FALSE	—	Alarm Level of Alarm 3	User
	b3	ALML4	BOOL	FALSE	TRUE	FALSE	—	Alarm Level of Alarm 4	User
	b4	ALML5	BOOL	FALSE	TRUE	FALSE	—	Alarm Level of Alarm 5	User
	b5	ALML6	BOOL	FALSE	TRUE	FALSE	—	Alarm Level of Alarm 6	User
	b6	ALML7	BOOL	FALSE	TRUE	FALSE	—	Alarm Level of Alarm 7	User
	b7	ALML8	BOOL	FALSE	TRUE	FALSE	—	Alarm Level of Alarm 8	User
+4	ALM1NO	INT	0	10000	0	—	Alarm 1 Name No.	User	
+5	ALM2NO	INT	0	10000	0	—	Alarm 2 Name No.	User	
+6	ALM3NO	INT	0	10000	0	—	Alarm 3 Name No.	User	
+7	ALM4NO	INT	0	10000	0	—	Alarm 4 Name No.	User	
+8	ALM5NO	INT	0	10000	0	—	Alarm 5 Name No.	User	
+9	ALM6NO	INT	0	10000	0	—	Alarm 6 Name No.	User	
+10	ALM7NO	INT	0	10000	0	—	Alarm 7 Name No.	User	
+11	ALM8NO	INT	0	10000	0	—	Alarm 8 Name No.	User	

# ALM\_64PT

Structure name: M+TM\_ALM\_64PT

Offset	Label	Data type	Setting/Storage range		Initial value	Unit	Application	Storage	
			Low limit	High limit					
+0	FUNC	INT	257	257	257	—	Tag Function Code	System	
+1	b0	ALM1	BOOL	FALSE	TRUE	FALSE	—	Alarm 1 to 16: Alarm 1	System
	b1	ALM2	BOOL	FALSE	TRUE	FALSE	—	Alarm 1 to 16: Alarm 2	System
	b2	ALM3	BOOL	FALSE	TRUE	FALSE	—	Alarm 1 to 16: Alarm 3	System
	b3	ALM4	BOOL	FALSE	TRUE	FALSE	—	Alarm 1 to 16: Alarm 4	System
	b4	ALM5	BOOL	FALSE	TRUE	FALSE	—	Alarm 1 to 16: Alarm 5	System
	b5	ALM6	BOOL	FALSE	TRUE	FALSE	—	Alarm 1 to 16: Alarm 6	System
	b6	ALM7	BOOL	FALSE	TRUE	FALSE	—	Alarm 1 to 16: Alarm 7	System
	b7	ALM8	BOOL	FALSE	TRUE	FALSE	—	Alarm 1 to 16: Alarm 8	System
	b8	ALM9	BOOL	FALSE	TRUE	FALSE	—	Alarm 1 to 16: Alarm 9	System
	b9	ALM10	BOOL	FALSE	TRUE	FALSE	—	Alarm 1 to 16: Alarm 10	System
	bA	ALM11	BOOL	FALSE	TRUE	FALSE	—	Alarm 1 to 16: Alarm 11	System
	bB	ALM12	BOOL	FALSE	TRUE	FALSE	—	Alarm 1 to 16: Alarm 12	System
	bC	ALM13	BOOL	FALSE	TRUE	FALSE	—	Alarm 1 to 16: Alarm 13	System
	bD	ALM14	BOOL	FALSE	TRUE	FALSE	—	Alarm 1 to 16: Alarm 14	System
	bE	ALM15	BOOL	FALSE	TRUE	FALSE	—	Alarm 1 to 16: Alarm 15	System
	bF	ALM16	BOOL	FALSE	TRUE	FALSE	—	Alarm 1 to 16: Alarm 16	System
+2	b0	ALM17	BOOL	FALSE	TRUE	FALSE	—	Alarm 17 to 32: Alarm 17	System
	b1	ALM18	BOOL	FALSE	TRUE	FALSE	—	Alarm 17 to 32: Alarm 18	System
	b2	ALM19	BOOL	FALSE	TRUE	FALSE	—	Alarm 17 to 32: Alarm 19	System
	b3	ALM20	BOOL	FALSE	TRUE	FALSE	—	Alarm 17 to 32: Alarm 20	System
	b4	ALM21	BOOL	FALSE	TRUE	FALSE	—	Alarm 17 to 32: Alarm 21	System
	b5	ALM22	BOOL	FALSE	TRUE	FALSE	—	Alarm 17 to 32: Alarm 22	System
	b6	ALM23	BOOL	FALSE	TRUE	FALSE	—	Alarm 17 to 32: Alarm 23	System
	b7	ALM24	BOOL	FALSE	TRUE	FALSE	—	Alarm 17 to 32: Alarm 24	System
	b8	ALM25	BOOL	FALSE	TRUE	FALSE	—	Alarm 17 to 32: Alarm 25	System
	b9	ALM26	BOOL	FALSE	TRUE	FALSE	—	Alarm 17 to 32: Alarm 26	System
	bA	ALM27	BOOL	FALSE	TRUE	FALSE	—	Alarm 17 to 32: Alarm 27	System
	bB	ALM28	BOOL	FALSE	TRUE	FALSE	—	Alarm 17 to 32: Alarm 28	System
	bC	ALM29	BOOL	FALSE	TRUE	FALSE	—	Alarm 17 to 32: Alarm 29	System
	bD	ALM30	BOOL	FALSE	TRUE	FALSE	—	Alarm 17 to 32: Alarm 30	System
	bE	ALM31	BOOL	FALSE	TRUE	FALSE	—	Alarm 17 to 32: Alarm 31	System
	bF	ALM32	BOOL	FALSE	TRUE	FALSE	—	Alarm 17 to 32: Alarm 32	System

Offset	Label	Data type	Setting/Storage range		Initial value	Unit	Application	Storage	
			Low limit	High limit					
+3	b0	ALM33	BOOL	FALSE	TRUE	FALSE	—	Alarm 33 to 48: Alarm 33	System
	b1	ALM34	BOOL	FALSE	TRUE	FALSE	—	Alarm 33 to 48: Alarm 34	System
	b2	ALM35	BOOL	FALSE	TRUE	FALSE	—	Alarm 33 to 48: Alarm 35	System
	b3	ALM36	BOOL	FALSE	TRUE	FALSE	—	Alarm 33 to 48: Alarm 36	System
	b4	ALM37	BOOL	FALSE	TRUE	FALSE	—	Alarm 33 to 48: Alarm 37	System
	b5	ALM38	BOOL	FALSE	TRUE	FALSE	—	Alarm 33 to 48: Alarm 38	System
	b6	ALM39	BOOL	FALSE	TRUE	FALSE	—	Alarm 33 to 48: Alarm 39	System
	b7	ALM40	BOOL	FALSE	TRUE	FALSE	—	Alarm 33 to 48: Alarm 40	System
	b8	ALM41	BOOL	FALSE	TRUE	FALSE	—	Alarm 33 to 48: Alarm 41	System
	b9	ALM42	BOOL	FALSE	TRUE	FALSE	—	Alarm 33 to 48: Alarm 42	System
	bA	ALM43	BOOL	FALSE	TRUE	FALSE	—	Alarm 33 to 48: Alarm 43	System
	bB	ALM44	BOOL	FALSE	TRUE	FALSE	—	Alarm 33 to 48: Alarm 44	System
	bC	ALM45	BOOL	FALSE	TRUE	FALSE	—	Alarm 33 to 48: Alarm 45	System
	bD	ALM46	BOOL	FALSE	TRUE	FALSE	—	Alarm 33 to 48: Alarm 46	System
	bE	ALM47	BOOL	FALSE	TRUE	FALSE	—	Alarm 33 to 48: Alarm 47	System
	bF	ALM48	BOOL	FALSE	TRUE	FALSE	—	Alarm 33 to 48: Alarm 48	System
+4	b0	ALM49	BOOL	FALSE	TRUE	FALSE	—	Alarm 49 to 64: Alarm 49	System
	b1	ALM50	BOOL	FALSE	TRUE	FALSE	—	Alarm 49 to 64: Alarm 50	System
	b2	ALM51	BOOL	FALSE	TRUE	FALSE	—	Alarm 49 to 64: Alarm 51	System
	b3	ALM52	BOOL	FALSE	TRUE	FALSE	—	Alarm 49 to 64: Alarm 52	System
	b4	ALM53	BOOL	FALSE	TRUE	FALSE	—	Alarm 49 to 64: Alarm 53	System
	b5	ALM54	BOOL	FALSE	TRUE	FALSE	—	Alarm 49 to 64: Alarm 54	System
	b6	ALM55	BOOL	FALSE	TRUE	FALSE	—	Alarm 49 to 64: Alarm 55	System
	b7	ALM56	BOOL	FALSE	TRUE	FALSE	—	Alarm 49 to 64: Alarm 56	System
	b8	ALM57	BOOL	FALSE	TRUE	FALSE	—	Alarm 49 to 64: Alarm 57	System
	b9	ALM58	BOOL	FALSE	TRUE	FALSE	—	Alarm 49 to 64: Alarm 58	System
	bA	ALM59	BOOL	FALSE	TRUE	FALSE	—	Alarm 49 to 64: Alarm 59	System
	bB	ALM60	BOOL	FALSE	TRUE	FALSE	—	Alarm 49 to 64: Alarm 60	System
	bC	ALM61	BOOL	FALSE	TRUE	FALSE	—	Alarm 49 to 64: Alarm 61	System
	bD	ALM62	BOOL	FALSE	TRUE	FALSE	—	Alarm 49 to 64: Alarm 62	System
	bE	ALM63	BOOL	FALSE	TRUE	FALSE	—	Alarm 49 to 64: Alarm 63	System
	bF	ALM64	BOOL	FALSE	TRUE	FALSE	—	Alarm 49 to 64: Alarm 64	System



Offset	Label	Data type	Setting/Storage range		Initial value	Unit	Application	Storage	
			Low limit	High limit					
+5	b0	ALML1	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 1 to 16: Alarm Level of Alarm 1	User
	b1	ALML2	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 1 to 16: Alarm Level of Alarm 2	User
	b2	ALML3	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 1 to 16: Alarm Level of Alarm 3	User
	b3	ALML4	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 1 to 16: Alarm Level of Alarm 4	User
	b4	ALML5	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 1 to 16: Alarm Level of Alarm 5	User
	b5	ALML6	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 1 to 16: Alarm Level of Alarm 6	User
	b6	ALML7	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 1 to 16: Alarm Level of Alarm 7	User
	b7	ALML8	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 1 to 16: Alarm Level of Alarm 8	User
	b8	ALML9	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 1 to 16: Alarm Level of Alarm 9	User
	b9	ALML10	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 1 to 16: Alarm Level of Alarm 10	User
	bA	ALML11	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 1 to 16: Alarm Level of Alarm 11	User
	bB	ALML12	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 1 to 16: Alarm Level of Alarm 12	User
	bC	ALML13	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 1 to 16: Alarm Level of Alarm 13	User
	bD	ALML14	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 1 to 16: Alarm Level of Alarm 14	User
	bE	ALML15	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 1 to 16: Alarm Level of Alarm 15	User
	bF	ALML16	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 1 to 16: Alarm Level of Alarm 16	User



Offset	Label	Data type	Setting/Storage range		Initial value	Unit	Application	Storage	
			Low limit	High limit					
+6	b0	ALML17	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 17 to 32: Alarm Level of Alarm 17	User
	b1	ALML18	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 17 to 32: Alarm Level of Alarm 18	User
	b2	ALML19	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 17 to 32: Alarm Level of Alarm 19	User
	b3	ALML20	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 17 to 32: Alarm Level of Alarm 20	User
	b4	ALML21	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 17 to 32: Alarm Level of Alarm 21	User
	b5	ALML22	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 17 to 32: Alarm Level of Alarm 22	User
	b6	ALML23	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 17 to 32: Alarm Level of Alarm 23	User
	b7	ALML24	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 17 to 32: Alarm Level of Alarm 24	User
	b8	ALML25	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 17 to 32: Alarm Level of Alarm 25	User
	b9	ALML26	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 17 to 32: Alarm Level of Alarm 26	User
	bA	ALML27	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 17 to 32: Alarm Level of Alarm 27	User
	bB	ALML28	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 17 to 32: Alarm Level of Alarm 28	User
	bC	ALML29	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 17 to 32: Alarm Level of Alarm 29	User
	bD	ALML30	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 17 to 32: Alarm Level of Alarm 30	User
	bE	ALML31	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 17 to 32: Alarm Level of Alarm 31	User
	bF	ALML32	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 17 to 32: Alarm Level of Alarm 32	User



Offset	Label	Data type	Setting/Storage range		Initial value	Unit	Application	Storage	
			Low limit	High limit					
+7	b0	ALML33	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 33 to 48: Alarm Level of Alarm 33	User
	b1	ALML34	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 33 to 48: Alarm Level of Alarm 34	User
	b2	ALML35	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 33 to 48: Alarm Level of Alarm 35	User
	b3	ALML36	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 33 to 48: Alarm Level of Alarm 36	User
	b4	ALML37	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 33 to 48: Alarm Level of Alarm 37	User
	b5	ALML38	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 33 to 48: Alarm Level of Alarm 38	User
	b6	ALML39	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 33 to 48: Alarm Level of Alarm 39	User
	b7	ALML40	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 33 to 48: Alarm Level of Alarm 40	User
	b8	ALML41	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 33 to 48: Alarm Level of Alarm 41	User
	b9	ALML42	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 33 to 48: Alarm Level of Alarm 42	User
	bA	ALML43	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 33 to 48: Alarm Level of Alarm 43	User
	bB	ALML44	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 33 to 48: Alarm Level of Alarm 44	User
	bC	ALML45	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 33 to 48: Alarm Level of Alarm 45	User
	bD	ALML46	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 33 to 48: Alarm Level of Alarm 46	User
	bE	ALML47	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 33 to 48: Alarm Level of Alarm 47	User
	bF	ALML48	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 33 to 48: Alarm Level of Alarm 48	User

Offset	Label	Data type	Setting/Storage range		Initial value	Unit	Application	Storage	
			Low limit	High limit					
+8	b0	ALML49	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 49 to 64: Alarm Level of Alarm 49	User
	b1	ALML50	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 49 to 64: Alarm Level of Alarm 50	User
	b2	ALML51	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 49 to 64: Alarm Level of Alarm 51	User
	b3	ALML52	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 49 to 64: Alarm Level of Alarm 52	User
	b4	ALML53	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 49 to 64: Alarm Level of Alarm 53	User
	b5	ALML54	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 49 to 64: Alarm Level of Alarm 54	User
	b6	ALML55	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 49 to 64: Alarm Level of Alarm 55	User
	b7	ALML56	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 49 to 64: Alarm Level of Alarm 56	User
	b8	ALML57	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 49 to 64: Alarm Level of Alarm 57	User
	b9	ALML58	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 49 to 64: Alarm Level of Alarm 58	User
	bA	ALML59	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 49 to 64: Alarm Level of Alarm 59	User
	bB	ALML60	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 49 to 64: Alarm Level of Alarm 60	User
	bC	ALML61	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 49 to 64: Alarm Level of Alarm 61	User
	bD	ALML62	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 49 to 64: Alarm Level of Alarm 62	User
	bE	ALML63	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 49 to 64: Alarm Level of Alarm 63	User
	bF	ALML64	BOOL	FALSE	TRUE	FALSE	—	Alarm Level 49 to 64: Alarm Level of Alarm 64	User
+9	ALM1NO	INT	0	10000	0	—	Alarm 1 Name No.	User	
+10	ALM2NO	INT	0	10000	0	—	Alarm 2 Name No.	User	
+11	ALM3NO	INT	0	10000	0	—	Alarm 3 Name No.	User	
+12	ALM4NO	INT	0	10000	0	—	Alarm 4 Name No.	User	
+13	ALM5NO	INT	0	10000	0	—	Alarm 5 Name No.	User	
+14	ALM6NO	INT	0	10000	0	—	Alarm 6 Name No.	User	
+15	ALM7NO	INT	0	10000	0	—	Alarm 7 Name No.	User	
+16	ALM8NO	INT	0	10000	0	—	Alarm 8 Name No.	User	
+17	ALM9NO	INT	0	10000	0	—	Alarm 9 Name No.	User	
+18	ALM10NO	INT	0	10000	0	—	Alarm 10 Name No.	User	
+19	ALM11NO	INT	0	10000	0	—	Alarm 11 Name No.	User	
+20	ALM12NO	INT	0	10000	0	—	Alarm 12 Name No.	User	
+21	ALM13NO	INT	0	10000	0	—	Alarm 13 Name No.	User	
+22	ALM14NO	INT	0	10000	0	—	Alarm 14 Name No.	User	
+23	ALM15NO	INT	0	10000	0	—	Alarm 15 Name No.	User	
+24	ALM16NO	INT	0	10000	0	—	Alarm 16 Name No.	User	
+25	ALM17NO	INT	0	10000	0	—	Alarm 17 Name No.	User	
+26	ALM18NO	INT	0	10000	0	—	Alarm 18 Name No.	User	
+27	ALM19NO	INT	0	10000	0	—	Alarm 19 Name No.	User	
+28	ALM20NO	INT	0	10000	0	—	Alarm 20 Name No.	User	
+29	ALM21NO	INT	0	10000	0	—	Alarm 21 Name No.	User	
+30	ALM22NO	INT	0	10000	0	—	Alarm 22 Name No.	User	
+31	ALM23NO	INT	0	10000	0	—	Alarm 23 Name No.	User	
+32	ALM24NO	INT	0	10000	0	—	Alarm 24 Name No.	User	



Offset	Label	Data type	Setting/Storage range		Initial value	Unit	Application	Storage
			Low limit	High limit				
+33	ALM25NO	INT	0	10000	0	—	Alarm 25 Name No.	User
+34	ALM26NO	INT	0	10000	0	—	Alarm 26 Name No.	User
+35	ALM27NO	INT	0	10000	0	—	Alarm 27 Name No.	User
+36	ALM28NO	INT	0	10000	0	—	Alarm 28 Name No.	User
+37	ALM29NO	INT	0	10000	0	—	Alarm 29 Name No.	User
+38	ALM30NO	INT	0	10000	0	—	Alarm 30 Name No.	User
+39	ALM31NO	INT	0	10000	0	—	Alarm 31 Name No.	User
+40	ALM32NO	INT	0	10000	0	—	Alarm 32 Name No.	User
+41	ALM33NO	INT	0	10000	0	—	Alarm 33 Name No.	User
+42	ALM34NO	INT	0	10000	0	—	Alarm 34 Name No.	User
+43	ALM35NO	INT	0	10000	0	—	Alarm 35 Name No.	User
+44	ALM36NO	INT	0	10000	0	—	Alarm 36 Name No.	User
+45	ALM37NO	INT	0	10000	0	—	Alarm 37 Name No.	User
+46	ALM38NO	INT	0	10000	0	—	Alarm 38 Name No.	User
+47	ALM39NO	INT	0	10000	0	—	Alarm 39 Name No.	User
+48	ALM40NO	INT	0	10000	0	—	Alarm 40 Name No.	User
+49	ALM41NO	INT	0	10000	0	—	Alarm 41 Name No.	User
+50	ALM42NO	INT	0	10000	0	—	Alarm 42 Name No.	User
+51	ALM43NO	INT	0	10000	0	—	Alarm 43 Name No.	User
+52	ALM44NO	INT	0	10000	0	—	Alarm 44 Name No.	User
+53	ALM45NO	INT	0	10000	0	—	Alarm 45 Name No.	User
+54	ALM46NO	INT	0	10000	0	—	Alarm 46 Name No.	User
+55	ALM47NO	INT	0	10000	0	—	Alarm 47 Name No.	User
+56	ALM48NO	INT	0	10000	0	—	Alarm 48 Name No.	User
+57	ALM49NO	INT	0	10000	0	—	Alarm 49 Name No.	User
+58	ALM50NO	INT	0	10000	0	—	Alarm 50 Name No.	User
+59	ALM51NO	INT	0	10000	0	—	Alarm 51 Name No.	User
+60	ALM52NO	INT	0	10000	0	—	Alarm 52 Name No.	User
+61	ALM53NO	INT	0	10000	0	—	Alarm 53 Name No.	User
+62	ALM54NO	INT	0	10000	0	—	Alarm 54 Name No.	User
+63	ALM55NO	INT	0	10000	0	—	Alarm 55 Name No.	User
+64	ALM56NO	INT	0	10000	0	—	Alarm 56 Name No.	User
+65	ALM57NO	INT	0	10000	0	—	Alarm 57 Name No.	User
+66	ALM58NO	INT	0	10000	0	—	Alarm 58 Name No.	User
+67	ALM59NO	INT	0	10000	0	—	Alarm 59 Name No.	User
+68	ALM60NO	INT	0	10000	0	—	Alarm 60 Name No.	User
+69	ALM61NO	INT	0	10000	0	—	Alarm 61 Name No.	User
+70	ALM62NO	INT	0	10000	0	—	Alarm 62 Name No.	User
+71	ALM63NO	INT	0	10000	0	—	Alarm 63 Name No.	User
+72	ALM64NO	INT	0	10000	0	—	Alarm 64 Name No.	User
+73	ALMG1NO	INT	0	10000	0	—	Alarm Group 1 Name No.	User
+74	ALMG2NO	INT	0	10000	0	—	Alarm Group 2 Name No.	User
+75	ALMG3NO	INT	0	10000	0	—	Alarm Group 3 Name No.	User
+76	ALMG4NO	INT	0	10000	0	—	Alarm Group 4 Name No.	User
+77	ALMG5NO	INT	0	10000	0	—	Alarm Group 5 Name No.	User
+78	ALMG6NO	INT	0	10000	0	—	Alarm Group 6 Name No.	User
+79	ALMG7NO	INT	0	10000	0	—	Alarm Group 7 Name No.	User
+80	ALMG8NO	INT	0	10000	0	—	Alarm Group 8 Name No.	User

# MSG

Structure name: M+TM\_MSG

Offset	Label	Data type	Setting/Storage range		Initial value	Unit	Application	Storage	
			Low limit	High limit					
+0	FUNC	INT	272	272	272	—	Tag Function Code	System	
+1	b0	MSG1	BOOL	FALSE	TRUE	FALSE	—	Message: Message 1	System
	b1	MSG2	BOOL	FALSE	TRUE	FALSE	—	Message: Message 2	System
	b2	MSG3	BOOL	FALSE	TRUE	FALSE	—	Message: Message 3	System
	b3	MSG4	BOOL	FALSE	TRUE	FALSE	—	Message: Message 4	System
	b4	MSG5	BOOL	FALSE	TRUE	FALSE	—	Message: Message 5	System
	b5	MSG6	BOOL	FALSE	TRUE	FALSE	—	Message: Message 6	System
	b6	MSG7	BOOL	FALSE	TRUE	FALSE	—	Message: Message 7	System
	b7	MSG8	BOOL	FALSE	TRUE	FALSE	—	Message: Message 8	System
+2	b0	MSGCHK1	BOOL	FALSE	TRUE	FALSE	—	Message Check: Message Check 1	User
	b1	MSGCHK2	BOOL	FALSE	TRUE	FALSE	—	Message Check: Message Check 2	User
	b2	MSGCHK3	BOOL	FALSE	TRUE	FALSE	—	Message Check: Message Check 3	User
	b3	MSGCHK4	BOOL	FALSE	TRUE	FALSE	—	Message Check: Message Check 4	User
	b4	MSGCHK5	BOOL	FALSE	TRUE	FALSE	—	Message Check: Message Check 5	User
	b5	MSGCHK6	BOOL	FALSE	TRUE	FALSE	—	Message Check: Message Check 6	User
	b6	MSGCHK7	BOOL	FALSE	TRUE	FALSE	—	Message Check: Message Check 7	User
	b7	MSGCHK8	BOOL	FALSE	TRUE	FALSE	—	Message Check: Message Check 8	User
+4	MSG1NO	INT	0	10000	0	—	Message 1 Name No.	User	
+5	MSG2NO	INT	0	10000	0	—	Message 2 Name No.	User	
+6	MSG3NO	INT	0	10000	0	—	Message 3 Name No.	User	
+7	MSG4NO	INT	0	10000	0	—	Message 4 Name No.	User	
+8	MSG5NO	INT	0	10000	0	—	Message 5 Name No.	User	
+9	MSG6NO	INT	0	10000	0	—	Message 6 Name No.	User	
+10	MSG7NO	INT	0	10000	0	—	Message 7 Name No.	User	
+11	MSG8NO	INT	0	10000	0	—	Message 8 Name No.	User	



# MSG\_64PT

Structure name: M+TM\_MSG\_64PT

Offset	Label	Data type	Setting/Storage range		Initial value	Unit	Application	Storage	
			Low limit	High limit					
+0	FUNC	INT	273	273	273	—	Tag Function Code	System	
+1	b0	MSG1	BOOL	FALSE	TRUE	FALSE	—	Message 1 to 16: Message 1	System
	b1	MSG2	BOOL	FALSE	TRUE	FALSE	—	Message 1 to 16: Message 2	System
	b2	MSG3	BOOL	FALSE	TRUE	FALSE	—	Message 1 to 16: Message 3	System
	b3	MSG4	BOOL	FALSE	TRUE	FALSE	—	Message 1 to 16: Message 4	System
	b4	MSG5	BOOL	FALSE	TRUE	FALSE	—	Message 1 to 16: Message 5	System
	b5	MSG6	BOOL	FALSE	TRUE	FALSE	—	Message 1 to 16: Message 6	System
	b6	MSG7	BOOL	FALSE	TRUE	FALSE	—	Message 1 to 16: Message 7	System
	b7	MSG8	BOOL	FALSE	TRUE	FALSE	—	Message 1 to 16: Message 8	System
	b8	MSG9	BOOL	FALSE	TRUE	FALSE	—	Message 1 to 16: Message 9	System
	b9	MSG10	BOOL	FALSE	TRUE	FALSE	—	Message 1 to 16: Message 10	System
	bA	MSG11	BOOL	FALSE	TRUE	FALSE	—	Message 1 to 16: Message 11	System
	bB	MSG12	BOOL	FALSE	TRUE	FALSE	—	Message 1 to 16: Message 12	System
	bC	MSG13	BOOL	FALSE	TRUE	FALSE	—	Message 1 to 16: Message 13	System
	bD	MSG14	BOOL	FALSE	TRUE	FALSE	—	Message 1 to 16: Message 14	System
	bE	MSG15	BOOL	FALSE	TRUE	FALSE	—	Message 1 to 16: Message 15	System
	bF	MSG16	BOOL	FALSE	TRUE	FALSE	—	Message 1 to 16: Message 16	System
+2	b0	MSG17	BOOL	FALSE	TRUE	FALSE	—	Message 17 to 32: Message 17	System
	b1	MSG18	BOOL	FALSE	TRUE	FALSE	—	Message 17 to 32: Message 18	System
	b2	MSG19	BOOL	FALSE	TRUE	FALSE	—	Message 17 to 32: Message 19	System
	b3	MSG20	BOOL	FALSE	TRUE	FALSE	—	Message 17 to 32: Message 20	System
	b4	MSG21	BOOL	FALSE	TRUE	FALSE	—	Message 17 to 32: Message 21	System
	b5	MSG22	BOOL	FALSE	TRUE	FALSE	—	Message 17 to 32: Message 22	System
	b6	MSG23	BOOL	FALSE	TRUE	FALSE	—	Message 17 to 32: Message 23	System
	b7	MSG24	BOOL	FALSE	TRUE	FALSE	—	Message 17 to 32: Message 24	System
	b8	MSG25	BOOL	FALSE	TRUE	FALSE	—	Message 17 to 32: Message 25	System
	b9	MSG26	BOOL	FALSE	TRUE	FALSE	—	Message 17 to 32: Message 26	System
	bA	MSG27	BOOL	FALSE	TRUE	FALSE	—	Message 17 to 32: Message 27	System
	bB	MSG28	BOOL	FALSE	TRUE	FALSE	—	Message 17 to 32: Message 28	System
	bC	MSG29	BOOL	FALSE	TRUE	FALSE	—	Message 17 to 32: Message 29	System
	bD	MSG30	BOOL	FALSE	TRUE	FALSE	—	Message 17 to 32: Message 30	System
	bE	MSG31	BOOL	FALSE	TRUE	FALSE	—	Message 17 to 32: Message 31	System
	bF	MSG32	BOOL	FALSE	TRUE	FALSE	—	Message 17 to 32: Message 32	System

Offset	Label	Data type	Setting/Storage range		Initial value	Unit	Application	Storage	
			Low limit	High limit					
+3	b0	MSG33	BOOL	FALSE	TRUE	FALSE	—	Message 33 to 48: Message 33	System
	b1	MSG34	BOOL	FALSE	TRUE	FALSE	—	Message 33 to 48: Message 34	System
	b2	MSG35	BOOL	FALSE	TRUE	FALSE	—	Message 33 to 48: Message 35	System
	b3	MSG36	BOOL	FALSE	TRUE	FALSE	—	Message 33 to 48: Message 36	System
	b4	MSG37	BOOL	FALSE	TRUE	FALSE	—	Message 33 to 48: Message 37	System
	b5	MSG38	BOOL	FALSE	TRUE	FALSE	—	Message 33 to 48: Message 38	System
	b6	MSG39	BOOL	FALSE	TRUE	FALSE	—	Message 33 to 48: Message 39	System
	b7	MSG40	BOOL	FALSE	TRUE	FALSE	—	Message 33 to 48: Message 40	System
	b8	MSG41	BOOL	FALSE	TRUE	FALSE	—	Message 33 to 48: Message 41	System
	b9	MSG42	BOOL	FALSE	TRUE	FALSE	—	Message 33 to 48: Message 42	System
	bA	MSG43	BOOL	FALSE	TRUE	FALSE	—	Message 33 to 48: Message 43	System
	bB	MSG44	BOOL	FALSE	TRUE	FALSE	—	Message 33 to 48: Message 44	System
	bC	MSG45	BOOL	FALSE	TRUE	FALSE	—	Message 33 to 48: Message 45	System
	bD	MSG46	BOOL	FALSE	TRUE	FALSE	—	Message 33 to 48: Message 46	System
	bE	MSG47	BOOL	FALSE	TRUE	FALSE	—	Message 33 to 48: Message 47	System
	bF	MSG48	BOOL	FALSE	TRUE	FALSE	—	Message 33 to 48: Message 48	System
+4	b0	MSG49	BOOL	FALSE	TRUE	FALSE	—	Message 49 to 64: Message 49	System
	b1	MSG50	BOOL	FALSE	TRUE	FALSE	—	Message 49 to 64: Message 50	System
	b2	MSG51	BOOL	FALSE	TRUE	FALSE	—	Message 49 to 64: Message 51	System
	b3	MSG52	BOOL	FALSE	TRUE	FALSE	—	Message 49 to 64: Message 52	System
	b4	MSG53	BOOL	FALSE	TRUE	FALSE	—	Message 49 to 64: Message 53	System
	b5	MSG54	BOOL	FALSE	TRUE	FALSE	—	Message 49 to 64: Message 54	System
	b6	MSG55	BOOL	FALSE	TRUE	FALSE	—	Message 49 to 64: Message 55	System
	b7	MSG56	BOOL	FALSE	TRUE	FALSE	—	Message 49 to 64: Message 56	System
	b8	MSG57	BOOL	FALSE	TRUE	FALSE	—	Message 49 to 64: Message 57	System
	b9	MSG58	BOOL	FALSE	TRUE	FALSE	—	Message 49 to 64: Message 58	System
	bA	MSG59	BOOL	FALSE	TRUE	FALSE	—	Message 49 to 64: Message 59	System
	bB	MSG60	BOOL	FALSE	TRUE	FALSE	—	Message 49 to 64: Message 60	System
	bC	MSG61	BOOL	FALSE	TRUE	FALSE	—	Message 49 to 64: Message 61	System
	bD	MSG62	BOOL	FALSE	TRUE	FALSE	—	Message 49 to 64: Message 62	System
	bE	MSG63	BOOL	FALSE	TRUE	FALSE	—	Message 49 to 64: Message 63	System
	bF	MSG64	BOOL	FALSE	TRUE	FALSE	—	Message 49 to 64: Message 64	System



Offset	Label	Data type	Setting/Storage range		Initial value	Unit	Application	Storage	
			Low limit	High limit					
+5	b0	MSGCHK1	BOOL	FALSE	TRUE	FALSE	—	Message Check 1 to 16: Message Check 1	User
	b1	MSGCHK2	BOOL	FALSE	TRUE	FALSE	—	Message Check 1 to 16: Message Check 2	User
	b2	MSGCHK3	BOOL	FALSE	TRUE	FALSE	—	Message Check 1 to 16: Message Check 3	User
	b3	MSGCHK4	BOOL	FALSE	TRUE	FALSE	—	Message Check 1 to 16: Message Check 4	User
	b4	MSGCHK5	BOOL	FALSE	TRUE	FALSE	—	Message Check 1 to 16: Message Check 5	User
	b5	MSGCHK6	BOOL	FALSE	TRUE	FALSE	—	Message Check 1 to 16: Message Check 6	User
	b6	MSGCHK7	BOOL	FALSE	TRUE	FALSE	—	Message Check 1 to 16: Message Check 7	User
	b7	MSGCHK8	BOOL	FALSE	TRUE	FALSE	—	Message Check 1 to 16: Message Check 8	User
	b8	MSGCHK9	BOOL	FALSE	TRUE	FALSE	—	Message Check 1 to 16: Message Check 9	User
	b9	MSGCHK10	BOOL	FALSE	TRUE	FALSE	—	Message Check 1 to 16: Message Check 10	User
	bA	MSGCHK11	BOOL	FALSE	TRUE	FALSE	—	Message Check 1 to 16: Message Check 11	User
	bB	MSGCHK12	BOOL	FALSE	TRUE	FALSE	—	Message Check 1 to 16: Message Check 12	User
	bC	MSGCHK13	BOOL	FALSE	TRUE	FALSE	—	Message Check 1 to 16: Message Check 13	User
	bD	MSGCHK14	BOOL	FALSE	TRUE	FALSE	—	Message Check 1 to 16: Message Check 14	User
	bE	MSGCHK15	BOOL	FALSE	TRUE	FALSE	—	Message Check 1 to 16: Message Check 15	User
	bF	MSGCHK16	BOOL	FALSE	TRUE	FALSE	—	Message Check 1 to 16: Message Check 16	User



Offset	Label	Data type	Setting/Storage range		Initial value	Unit	Application	Storage	
			Low limit	High limit					
+6	b0	MSGCHK17	BOOL	FALSE	TRUE	FALSE	—	Message Check 17 to 32: Message Check 17	User
	b1	MSGCHK18	BOOL	FALSE	TRUE	FALSE	—	Message Check 17 to 32: Message Check 18	User
	b2	MSGCHK19	BOOL	FALSE	TRUE	FALSE	—	Message Check 17 to 32: Message Check 19	User
	b3	MSGCHK20	BOOL	FALSE	TRUE	FALSE	—	Message Check 17 to 32: Message Check 20	User
	b4	MSGCHK21	BOOL	FALSE	TRUE	FALSE	—	Message Check 17 to 32: Message Check 21	User
	b5	MSGCHK22	BOOL	FALSE	TRUE	FALSE	—	Message Check 17 to 32: Message Check 22	User
	b6	MSGCHK23	BOOL	FALSE	TRUE	FALSE	—	Message Check 17 to 32: Message Check 23	User
	b7	MSGCHK24	BOOL	FALSE	TRUE	FALSE	—	Message Check 17 to 32: Message Check 24	User
	b8	MSGCHK25	BOOL	FALSE	TRUE	FALSE	—	Message Check 17 to 32: Message Check 25	User
	b9	MSGCHK26	BOOL	FALSE	TRUE	FALSE	—	Message Check 17 to 32: Message Check 26	User
	bA	MSGCHK27	BOOL	FALSE	TRUE	FALSE	—	Message Check 17 to 32: Message Check 27	User
	bB	MSGCHK28	BOOL	FALSE	TRUE	FALSE	—	Message Check 17 to 32: Message Check 28	User
	bC	MSGCHK29	BOOL	FALSE	TRUE	FALSE	—	Message Check 17 to 32: Message Check 29	User
	bD	MSGCHK30	BOOL	FALSE	TRUE	FALSE	—	Message Check 17 to 32: Message Check 30	User
	bE	MSGCHK31	BOOL	FALSE	TRUE	FALSE	—	Message Check 17 to 32: Message Check 31	User
	bF	MSGCHK32	BOOL	FALSE	TRUE	FALSE	—	Message Check 17 to 32: Message Check 32	User



Offset	Label	Data type	Setting/Storage range		Initial value	Unit	Application	Storage	
			Low limit	High limit					
+7	b0	MSGCHK33	BOOL	FALSE	TRUE	FALSE	—	Message Check 33 to 48: Message Check 33	User
	b1	MSGCHK34	BOOL	FALSE	TRUE	FALSE	—	Message Check 33 to 48: Message Check 34	User
	b2	MSGCHK35	BOOL	FALSE	TRUE	FALSE	—	Message Check 33 to 48: Message Check 35	User
	b3	MSGCHK36	BOOL	FALSE	TRUE	FALSE	—	Message Check 33 to 48: Message Check 36	User
	b4	MSGCHK37	BOOL	FALSE	TRUE	FALSE	—	Message Check 33 to 48: Message Check 37	User
	b5	MSGCHK38	BOOL	FALSE	TRUE	FALSE	—	Message Check 33 to 48: Message Check 38	User
	b6	MSGCHK39	BOOL	FALSE	TRUE	FALSE	—	Message Check 33 to 48: Message Check 39	User
	b7	MSGCHK40	BOOL	FALSE	TRUE	FALSE	—	Message Check 33 to 48: Message Check 40	User
	b8	MSGCHK41	BOOL	FALSE	TRUE	FALSE	—	Message Check 33 to 48: Message Check 41	User
	b9	MSGCHK42	BOOL	FALSE	TRUE	FALSE	—	Message Check 33 to 48: Message Check 42	User
	bA	MSGCHK43	BOOL	FALSE	TRUE	FALSE	—	Message Check 33 to 48: Message Check 43	User
	bB	MSGCHK44	BOOL	FALSE	TRUE	FALSE	—	Message Check 33 to 48: Message Check 44	User
	bC	MSGCHK45	BOOL	FALSE	TRUE	FALSE	—	Message Check 33 to 48: Message Check 45	User
	bD	MSGCHK46	BOOL	FALSE	TRUE	FALSE	—	Message Check 33 to 48: Message Check 46	User
	bE	MSGCHK47	BOOL	FALSE	TRUE	FALSE	—	Message Check 33 to 48: Message Check 47	User
	bF	MSGCHK48	BOOL	FALSE	TRUE	FALSE	—	Message Check 33 to 48: Message Check 48	User

Offset	Label	Data type	Setting/Storage range		Initial value	Unit	Application	Storage	
			Low limit	High limit					
+8	b0	MSGCHK49	BOOL	FALSE	TRUE	FALSE	—	Message Check 49 to 64: Message Check 49	User
	b1	MSGCHK50	BOOL	FALSE	TRUE	FALSE	—	Message Check 49 to 64: Message Check 50	User
	b2	MSGCHK51	BOOL	FALSE	TRUE	FALSE	—	Message Check 49 to 64: Message Check 51	User
	b3	MSGCHK52	BOOL	FALSE	TRUE	FALSE	—	Message Check 49 to 64: Message Check 52	User
	b4	MSGCHK53	BOOL	FALSE	TRUE	FALSE	—	Message Check 49 to 64: Message Check 53	User
	b5	MSGCHK54	BOOL	FALSE	TRUE	FALSE	—	Message Check 49 to 64: Message Check 54	User
	b6	MSGCHK55	BOOL	FALSE	TRUE	FALSE	—	Message Check 49 to 64: Message Check 55	User
	b7	MSGCHK56	BOOL	FALSE	TRUE	FALSE	—	Message Check 49 to 64: Message Check 56	User
	b8	MSGCHK57	BOOL	FALSE	TRUE	FALSE	—	Message Check 49 to 64: Message Check 57	User
	b9	MSGCHK58	BOOL	FALSE	TRUE	FALSE	—	Message Check 49 to 64: Message Check 58	User
	bA	MSGCHK59	BOOL	FALSE	TRUE	FALSE	—	Message Check 49 to 64: Message Check 59	User
	bB	MSGCHK60	BOOL	FALSE	TRUE	FALSE	—	Message Check 49 to 64: Message Check 60	User
	bC	MSGCHK61	BOOL	FALSE	TRUE	FALSE	—	Message Check 49 to 64: Message Check 61	User
	bD	MSGCHK62	BOOL	FALSE	TRUE	FALSE	—	Message Check 49 to 64: Message Check 62	User
	bE	MSGCHK63	BOOL	FALSE	TRUE	FALSE	—	Message Check 49 to 64: Message Check 63	User
	bF	MSGCHK64	BOOL	FALSE	TRUE	FALSE	—	Message Check 49 to 64: Message Check 64	User
+9		MSG1NO	INT	0	10000	0	—	Message 1 Name No.	User
+10		MSG2NO	INT	0	10000	0	—	Message 2 Name No.	User
+11		MSG3NO	INT	0	10000	0	—	Message 3 Name No.	User
+12		MSG4NO	INT	0	10000	0	—	Message 4 Name No.	User
+13		MSG5NO	INT	0	10000	0	—	Message 5 Name No.	User
+14		MSG6NO	INT	0	10000	0	—	Message 6 Name No.	User
+15		MSG7NO	INT	0	10000	0	—	Message 7 Name No.	User
+16		MSG8NO	INT	0	10000	0	—	Message 8 Name No.	User
+17		MSG9NO	INT	0	10000	0	—	Message 9 Name No.	User
+18		MSG10NO	INT	0	10000	0	—	Message 10 Name No.	User
+19		MSG11NO	INT	0	10000	0	—	Message 11 Name No.	User
+20		MSG12NO	INT	0	10000	0	—	Message 12 Name No.	User
+21		MSG13NO	INT	0	10000	0	—	Message 13 Name No.	User
+22		MSG14NO	INT	0	10000	0	—	Message 14 Name No.	User
+23		MSG15NO	INT	0	10000	0	—	Message 15 Name No.	User
+24		MSG16NO	INT	0	10000	0	—	Message 16 Name No.	User
+25		MSG17NO	INT	0	10000	0	—	Message 17 Name No.	User
+26		MSG18NO	INT	0	10000	0	—	Message 18 Name No.	User
+27		MSG19NO	INT	0	10000	0	—	Message 19 Name No.	User
+28		MSG20NO	INT	0	10000	0	—	Message 20 Name No.	User
+29		MSG21NO	INT	0	10000	0	—	Message 21 Name No.	User
+30		MSG22NO	INT	0	10000	0	—	Message 22 Name No.	User
+31		MSG23NO	INT	0	10000	0	—	Message 23 Name No.	User
+32		MSG24NO	INT	0	10000	0	—	Message 24 Name No.	User



Offset	Label	Data type	Setting/Storage range		Initial value	Unit	Application	Storage
			Low limit	High limit				
+33	MSG25NO	INT	0	10000	0	—	Message 25 Name No.	User
+34	MSG26NO	INT	0	10000	0	—	Message 26 Name No.	User
+35	MSG27NO	INT	0	10000	0	—	Message 27 Name No.	User
+36	MSG28NO	INT	0	10000	0	—	Message 28 Name No.	User
+37	MSG29NO	INT	0	10000	0	—	Message 29 Name No.	User
+38	MSG30NO	INT	0	10000	0	—	Message 30 Name No.	User
+39	MSG31NO	INT	0	10000	0	—	Message 31 Name No.	User
+40	MSG32NO	INT	0	10000	0	—	Message 32 Name No.	User
+41	MSG33NO	INT	0	10000	0	—	Message 33 Name No.	User
+42	MSG34NO	INT	0	10000	0	—	Message 34 Name No.	User
+43	MSG35NO	INT	0	10000	0	—	Message 35 Name No.	User
+44	MSG36NO	INT	0	10000	0	—	Message 36 Name No.	User
+45	MSG37NO	INT	0	10000	0	—	Message 37 Name No.	User
+46	MSG38NO	INT	0	10000	0	—	Message 38 Name No.	User
+47	MSG39NO	INT	0	10000	0	—	Message 39 Name No.	User
+48	MSG40NO	INT	0	10000	0	—	Message 40 Name No.	User
+49	MSG41NO	INT	0	10000	0	—	Message 41 Name No.	User
+50	MSG42NO	INT	0	10000	0	—	Message 42 Name No.	User
+51	MSG43NO	INT	0	10000	0	—	Message 43 Name No.	User
+52	MSG44NO	INT	0	10000	0	—	Message 44 Name No.	User
+53	MSG45NO	INT	0	10000	0	—	Message 45 Name No.	User
+54	MSG46NO	INT	0	10000	0	—	Message 46 Name No.	User
+55	MSG47NO	INT	0	10000	0	—	Message 47 Name No.	User
+56	MSG48NO	INT	0	10000	0	—	Message 48 Name No.	User
+57	MSG49NO	INT	0	10000	0	—	Message 49 Name No.	User
+58	MSG50NO	INT	0	10000	0	—	Message 50 Name No.	User
+59	MSG51NO	INT	0	10000	0	—	Message 51 Name No.	User
+60	MSG52NO	INT	0	10000	0	—	Message 52 Name No.	User
+61	MSG53NO	INT	0	10000	0	—	Message 53 Name No.	User
+62	MSG54NO	INT	0	10000	0	—	Message 54 Name No.	User
+63	MSG55NO	INT	0	10000	0	—	Message 55 Name No.	User
+64	MSG56NO	INT	0	10000	0	—	Message 56 Name No.	User
+65	MSG57NO	INT	0	10000	0	—	Message 57 Name No.	User
+66	MSG58NO	INT	0	10000	0	—	Message 58 Name No.	User
+67	MSG59NO	INT	0	10000	0	—	Message 59 Name No.	User
+68	MSG60NO	INT	0	10000	0	—	Message 60 Name No.	User
+69	MSG61NO	INT	0	10000	0	—	Message 61 Name No.	User
+70	MSG62NO	INT	0	10000	0	—	Message 62 Name No.	User
+71	MSG63NO	INT	0	10000	0	—	Message 63 Name No.	User
+72	MSG64NO	INT	0	10000	0	—	Message 64 Name No.	User
+73	MSGG1NO	INT	0	10000	0	—	Message Group 1 Name No.	User
+74	MSGG2NO	INT	0	10000	0	—	Message Group 2 Name No.	User
+75	MSGG3NO	INT	0	10000	0	—	Message Group 3 Name No.	User
+76	MSGG4NO	INT	0	10000	0	—	Message Group 4 Name No.	User
+77	MSGG5NO	INT	0	10000	0	—	Message Group 5 Name No.	User
+78	MSGG6NO	INT	0	10000	0	—	Message Group 6 Name No.	User
+79	MSGG7NO	INT	0	10000	0	—	Message Group 7 Name No.	User
+80	MSGG8NO	INT	0	10000	0	—	Message Group 8 Name No.	User

## Correspondence table of tag types and tag access FBs

The following table lists tag access FBs that can be used in user-defined tag FBs.

Classification	Tag type	Tag access FB
Loop tag	PID	M+P_IN, M+P_OUT1, M+P_DUTY, M+P_PID(_T), M+P_PHPL, M+P_MCHG
	2PID	M+P_IN, M+P_OUT1, M+P_DUTY, M+P_2PID(_T), M+P_PHPL, M+P_MCHG
	2PIDH	M+P_IN, M+P_OUT3, M+P_PHPL, M+P_2PIDH(_T), M+P_MCHG
	PIDP	M+P_IN, M+P_PIDP(_T), M+P_PIDP_EX(_T), M+P_PHPL, M+P_MCHG
	SPI	M+P_IN, M+P_OUT1, M+P_DUTY, M+P_SPI(_T), M+P_PHPL, M+P_MCHG
	IPD	M+P_IN, M+P_OUT1, M+P_DUTY, M+P_IPD(_T), M+P_PHPL, M+P_MCHG
	BPI	M+P_IN, M+P_OUT1, M+P_DUTY, M+P_BPI(_T), M+P_PHPL, M+P_MCHG
	R	M+P_IN, M+P_OUT2, M+P_R(_T), M+P_PHPL, M+P_MCHG
	ONF2	M+P_IN, M+P_PHPL, M+P_ONF2(_T), M+P_MCHG
	ONF3	M+P_IN, M+P_PHPL, M+P_ONF3(_T), M+P_MCHG
	MONI	M+P_IN, M+P_PHPL
	MWM	M+P_IN, M+P_MOUT, M+P_PHPL, M+P_MCHG
	BC	M+P_PSUM, M+P_BC
	PSUM	M+P_PSUM
	SEL	M+P_SEL(_T1) (_T2) (_T3), M+P_MCHG
	MOUT	M+P_MOUT, M+P_MCHG
	PGS	M+P_PGS, M+P_MCHG
	PGS2	M+P_PGS2, M+P_MCHG
	SWM	M+P_IN, M+P_MSET, M+P_PHPL, M+P_MCHG

## Correspondence table of tag types and tag FBs

The following table lists tag FBs that correspond to each of tag types.

Classification	Tag type	Tag FB
Loop tag	PID	M+M_PID(_T), M+M_PID_DUTY(_T)
	2PID	M+M_2PID(_T), M+M_2PID_DUTY(_T)
	2PIDH	M+M_2PIDH(_T)_
	PIDP	M+M_PIDP(_T), M+M_PIDP_EX(_T)_
	SPI	M+M_SPI(_T)
	IPD	M+M_IPD(_T)
	BPI	M+M_BPI(_T)
	R	M+M_R(_T)
	ONF2	M+M_ONF2(_T)
	ONF3	M+M_ONF3(_T)
	MONI	M+M_MONI
	MWM	M+M_MWM
	BC	M+M_BC
	PSUM	M+M_PSUM
	SEL	M+M_SEL(_T1)(_T2)(_T3_)
	MOUT	M+M_MOUT
	PGS	M+M_PGS
	PGS2	M+M_PGS2_
	SWM	M+M_SWM_
	PVAL	M+M_PVAL_T_
HTCL	M+M_HTCL_T_	
Status tag	NREV	M+M_NREV
	REV	M+M_REV
	MVAL1	M+M_MVAL1
	MVAL2	M+M_MVAL2
	TIMER1	M+M_TIMER1
	TIMER2	M+M_TIMER2
	COUNT1	M+M_COUNTER1
	COUNT2	M+M_COUNTER2
	PB	M+M_PB_
Alarm tag	ALM	M+M_ALARM
	ALM_64PT	M+M_ALARM_64PT_
Message tag	MSG	M+M_MESSAGE
	MSG_64PT	M+M_MESSAGE_64PT_

# Correspondence table of tag types and alarms

The following table lists alarms that correspond to each of tag types.

Classification	Tag type	Alarm
Loop tag	PID	SPA, DMLA, OOA, SEA, HHA, LLA, PHA, PLA, DPPA, DPNA, DVLA, MHA, MLA
	2PID	SPA, DMLA, OOA, SEA, HHA, LLA, PHA, PLA, DPPA, DPNA, DVLA, MHA, MLA
	2PIDH	SPA, DMLA, OOA, SEA, HHA, LLA, PHA, PLA, DPPA, DPNA, DVLA, MHA, MLA, SVHA, SVLA, DSVLA
	PIDP	SPA, DMLA, OOA, SEA, HHA, LLA, PHA, PLA, DPPA, DPNA, DVLA, MHA, MLA
	SPI	SPA, DMLA, OOA, SEA, HHA, LLA, PHA, PLA, DPPA, DPNA, DVLA, MHA, MLA
	IPD	SPA, DMLA, OOA, SEA, HHA, LLA, PHA, PLA, DPPA, DPNA, DVLA, MHA, MLA
	BPI	SPA, DMLA, OOA, SEA, HHA, LLA, PHA, PLA, DPPA, DPNA, DVLA, MHA, MLA
	R	SPA, DMLA, OOA, SEA, HHA, LLA, PHA, PLA, DPPA, DPNA, MHA, MLA
	ONF2	SPA, OOA, SEA, HHA, LLA, PHA, PLA, DPPA, DPNA
	ONF3	SPA, OOA, SEA, HHA, LLA, PHA, PLA, DPPA, DPNA
	MONI	SPA, SEA, HHA, LLA, PHA, PLA, DPPA
	MWM	SPA, OOA, SEA, HHA, LLA, PHA, PLA, DPPA
	BC	PHA, DPPA
	PSUM	—
	SEL	SPA, DMLA, OOA, MHA, MLA
	MOUT	SPA, OOA
	PGS	SPA, MHA, MLA
	PGS2	SPA, SVHA, SVLA
	SWM	SPA, OOA, SEA, HHA, LLA, PHA, PLA, DPPA, DPNA, DVLA, SVHA, SVLA, DSVLA
	PVAL	SPA, SEA, HHA, LLA, PHA, PLA, DPPA, DPNA, DVLA, TRIPA, TOA, SVHA, SVLA, DSVLA
HTCL	SPA, HBOA, OOA, DMLA_HT/CL, MHA_HT/CL, MLA_HT/CL	
Status tag	NREV	TRIPA, TOA
	REV	TRIPA, TOA
	MVAL1	TRIPA, TOA
	MVAL2	TRIPA, TOA
	TIMER1	—
	TIMER2	—
	COUNT1	—
	COUNT2	—
	PB	—



## Correspondence table of tag types and control modes

The following table lists control modes that correspond to each of tag types. (Only for the tag types that can switch the control mode)

Classification	Tag type	Control mode
Loop tag	PID	MAN, AUT, CAS, CMV, CSV
	2PID	MAN, AUT, CAS, CMV, CSV
	2PIDH	MAN, AUT, CAS, CMV, CSV, CASDR
	PIDP	MAN, AUT, CAS, CMV, CSV
	SPI	MAN, AUT, CAS, CMV, CSV
	IPD	MAN, AUT, CAS, CMV, CSV
	BPI	MAN, AUT, CAS, CMV, CSV
	R	MAN, AUT, CAS, CMV, CSV
	ONF2	MAN, AUT, CAS, CMV, CSV
	ONF3	MAN, AUT, CAS, CMV, CSV
	MONI	—
	MWM	MAN, CMV
	BC	—
	PSUM	—
	SEL	MAN, AUT, CAS, CMV, CSV
	MOUT	MAN, CMV
	PGS	MAN, AUT, CAS, CMV, CSV
	PGS2	MAN, AUT
	SWM	MAN, AUT, CAS, CSV
	PVAL	MAN, AUT, CAS, CSV
HTCL	MAN, AUT, CAS, CMV, CSV	
Status tag	NREV	MAN, AUT
	REV	MAN, AUT
	MVAL1	MAN, AUT
	MVAL2	MAN, AUT
	TIMER1	—
	TIMER2	—
	COUNT1	—
	COUNT2	—
	PB	MAN, AUT



## Correspondence table of tag types and I/O modes

The following table lists I/O modes that correspond to each of tag types. (Only for the tag types that can switch the I/O mode)

Classification	Tag type	I/O mode
Loop tag	PID	NOR, SIM, OVR, AT1
	2PID	NOR, SIM, OVR, AT1
	2PIDH	NOR, SIM, OVR, TSTP, AT1, AT2
	PIDP	NOR, SIM, OVR
	SPI	NOR, SIM, OVR
	IPD	NOR, SIM, OVR
	BPI	NOR, SIM, OVR
	R	NOR, SIM, OVR
	ONF2	NOR, OVR
	ONF3	NOR, OVR
	MONI	NOR, OVR
	MWM	NOR, OVR
	BC	—
	PSUM	—
	SEL	NOR
	MOUT	—
	PGS	—
	PGS2	NOR, TSTP
	SWM	NOR, OVR, TSTP
	PVAL	NOR, SIM, OVR, TSTP
HTCL	NOR, TSTP	
Status tag	NREV	NOR, SIM, OVR
	REV	NOR, SIM, OVR
	MVAL1	NOR, SIM, OVR
	MVAL2	NOR, SIM, OVR
	TIMER1	—
	TIMER2	—
	COUNT1	—
	COUNT2	—
	PB	—

# Appendix 2 Approximate Number of Steps

## List of number of steps

The following table lists the approximate number of steps of the process control function blocks.



For some function blocks, the approximate number of steps differs depending on their version.  
For how to check the version information of a function block, refer to the following.

Page 950 Version Upgrade

Item	Approximate number of steps
M+P_HS	50
M+P_HS_E	60
M+P_LS	50
M+P_LS_E	60
M+P_MID	50
M+P_MID_E	60
M+P_AVE	30
M+P_AVE_E	40
M+P_ABS	30
M+P_ABS_E	40
M+P_FG	30
M+P_IFG	30
M+P_FLT	40
M+P_ENG	40
M+P_IENG	40
M+P_TPC	80
M+P_SUM	50
M+P_SUM2_	280
M+P_RANGE_	160
M+P_ADD	50
M+P_SUB	50
M+P_MUL	50
M+P_DIV	90
M+P_SQR	40
M+P_GT	70
M+P_LT	70
M+P_EQ	50
M+P_GE	70
M+P_LE	70
M+P_LLAG	50
M+P_I	50
M+P_D	50
M+P_DED	120
M+P_LIMT	70
M+P_VLMT1	70
M+P_VLMT2	70
M+P_DBND	50
M+P_BUMP	60
M+P_AMR	70
M+P_DUTY_8PT_	1110
M+SR	20

Item	Approximate number of steps
M+RS	20
M+LATCH_BOOL	20
M+LATCH_REAL	20
M+LATCH_WORD	20
M+LATCH_DWORD	20
M+R_TRIG	20
M+F_TRIG	20
M+EDGE_CHECK	30
M+CTU	40
M+CTD	40
M+CTUD	70
M+TP_HIGH	90
M+TP_LOW	80
M+TON_HIGH	90
M+TON_LOW	80
M+TOF_HIGH	90
M+TOF_LOW	80
M+P_IN	Version lower than 1.070Y: 150 Version 1.070Y or later: 160 Version 1.085P or later: 180
M+P_OUT1	Version lower than 1.085P: 80 Version 1.085P or later: 100
M+P_OUT2	Version lower than 1.085P: 80 Version 1.085P or later: 90
M+P_OUT3_	Version lower than 1.042U: 950 Version 1.042U: 960 Version 1.057K or later: 970
M+P_MOUT	40
M+P_DUTY	Version lower than 1.085P: 50 Version 1.085P or later: 70
M+P_PSUM	180
M+P_BC	50
M+P_MSET_	750
M+P_R_T	50
M+P_R	40
M+P_PID_T	190
M+P_PID	180
M+P_2PID_T	190
M+P_2PID	180
M+P_2PIDH_T_	1810
M+P_2PIDH_	1770
M+P_PIDP_T	120
M+P_PIDP	110
M+P_PIDP_EX_T_	130
M+P_PIDP_EX_	120
M+P_SPI_T	70
M+P_SPI	60
M+P_IPD_T	70
M+P_IPD	70
M+P_BPI_T	120
M+P_BPI	120
M+P_PHPL	90
M+P_ONF2_T	70
M+P_ONF2	70
M+P_ONF3_T	70

Item	Approximate number of steps
M+P_ONF3	70
M+P_PGS	40
M+P_PGS2_	1760
M+P_SEL	70
M+P_SEL_T1	70
M+P_SEL_T2	80
M+P_SEL_T3_	590
M+P_MCHG	Version lower than 1.070Y: 160 Version 1.070Y or later: 180
M+P_MCHGPRMRY	100
M+M_PID_T	Version lower than 1.070Y: 130 Version 1.070Y or later: 160 Version 1.085P or later: 160
M+M_PID	130
M+M_PID_DUTY_T	Version lower than 1.070Y: 120 Version 1.070Y or later: 150
M+M_PID_DUTY	Version lower than 1.085P: 120 Version 1.085P or later: 130
M+M_2PID_T	Version lower than 1.070Y: 130 Version 1.070Y or later: 160
M+M_2PID	130
M+M_2PID_DUTY_T	Version lower than 1.070Y: 120 Version 1.070Y or later: 150
M+M_2PID_DUTY	Version lower than 1.085P: 120 Version 1.085P or later: 130
M+M_2PIDH_T_	Version lower than 1.070Y: 580 Version 1.070Y or later: 610 Version 1.085P or later: 630
M+M_2PIDH_	Version lower than 1.085P: 580 Version 1.085P or later: 600
M+M_PIDP_T	Version lower than 1.070Y: 110 Version 1.070Y or later: 140
M+M_PIDP	Version lower than 1.085P: 110 Version 1.085P or later: 120
M+M_PIDP_EX_T_	Version lower than 1.070Y: 120 Version 1.070Y or later: 140
M+M_PIDP_EX_	Version lower than 1.070Y: 110 Version 1.070Y or later: 120
M+M_SPI_T	Version lower than 1.070Y: 130 Version 1.070Y or later: 150 Version 1.085P or later: 160
M+M_SPI	Version lower than 1.070Y: 120 Version 1.070Y or later: 130
M+M_IPD_T	Version lower than 1.070Y: 130 Version 1.070Y or later: 160
M+M_IPD	130
M+M_BPI_T	Version lower than 1.070Y: 130 Version 1.070Y or later: 160
M+M_BPI	Version lower than 1.085P: 130 Version 1.085P or later: 140
M+M_R_T	Version lower than 1.070Y: 120 Version 1.070Y or later: 150
M+M_R	Version lower than 1.085P: 120 Version 1.085P or later: 130
M+M_ONF2_T	Version lower than 1.070Y: 100 Version 1.070Y or later: 130
M+M_ONF2	Version lower than 1.085P: 100 Version 1.085P or later: 110
M+M_ONF3_T	Version lower than 1.070Y: 110 Version 1.070Y or later: 130

Item	Approximate number of steps
M+M_ONF3	Version lower than 1.070Y: 100 Version 1.070Y or later: 110
M+M_MONI	60
M+M_MWM	Version lower than 1.085P: 90 Version 1.085P or later: 100
M+M_BC	90
M+M_PSUM	50
M+M_SEL	60
M+M_SEL_T1	Version lower than 1.070Y: 60 Version 1.070Y or later: 90
M+M_SEL_T2	Version lower than 1.070Y: 70 Version 1.070Y or later: 110
M+M_SEL_T3_	Version lower than 1.070Y: 70 Version 1.070Y or later: 110
M+M_MOUT	50
M+M_PGS	50
M+M_PGS2_	80
M+M_SWM_	Version lower than 1.070Y: 100 Version 1.070Y or later: 110
M+M_PVAL_T_	Version lower than 1.070Y: 1350 Version 1.070Y or later: 1380
M+M_HTCL_T_	Version lower than 1.070Y: 1380 Version 1.070Y or later: 1400
M+M_NREV	420
M+M_REV	540
M+M_MVAL1	440
M+M_MVAL2	530
M+M_TIMER1	360
M+M_TIMER2	350
M+M_COUNTER1	340
M+M_COUNTER2	330
M+M_PB_	240
M+M_ALARM	40
M+M_ALARM_64PT_	30
M+M_MESSAGE	40
M+M_MESSAGE_64PT_	30

## Increasing number of steps by the process control extension

When the process control extension is enabled, the number of steps increases comparing when it is not used.

The following table lists the increasing number of steps.

Execution type of program file	Program block	Increasing number of steps
Scan	M+PHEADER	Approximately 1400 steps
	M+PFOOTER	Approximately 30 steps
	Other than above	Approximately 10 steps
Fixed scan	All	Approximately 60 steps

# Appendix 3 Related Functions of Process

This section describes functions related to the process control function blocks.

## Auto tuning

The auto tuning function detects dynamic characteristics of a control target and automatically tunes the proportional gain (Kp), integral time (Ti), and derivative time (Td) for PID to suitable values.

The auto tuning function has two methods: step response method and limit cycle method.

Item	AT1 (Step response method)	AT2 (Limit cycle method)
Overview	In this method, a proportional gain (Kp), an integral time (Ti), and a derivative time (Td) for PID operations are calculated with the ZN method (Ziegler-Nichols' step response method) for setting their initial values.	In this method, a proportional gain (Kp), an integral time (Ti), and a derivative time (Td) for PID operations are calculated from the variation amplitude and variation cycle of a process variable obtained by repeatedly outputting the high and low limit values of a manipulated value. The limit cycle method is less influenced by the process variable noise compared to the step response method, leading a stable tuning result.
Applicable control mode	MANUAL, COMPUTER MV	AUTO, MANUAL, CASCADE, COMPUTER MV, COMPUTER SV
PID constants calculation specification method	<ul style="list-style-type: none"> <li>■P control tuning Execute tuning after setting Ti = 0 and Td = 0.</li> <li>■PI control tuning Execute tuning after setting Ti &gt; 0 and Td = 0.</li> <li>■PID control tuning Execute tuning after setting Ti &gt; 0 and Td &gt; 0.</li> </ul>	Execute tuning by selecting the PI control or PID control with the PX Developer monitor tool. (Tuning only for the P control cannot be executed.)
Corresponding tag access FB	<ul style="list-style-type: none"> <li>• M+P_PID(_T)</li> <li>• M+P_PID_DUTY(_T)</li> <li>• M+P_2PID(_T)</li> <li>• M+P_2PID_DUTY(_T)</li> <li>• M+P_2PIDH(_T)_</li> </ul>	<ul style="list-style-type: none"> <li>• M+P_2PIDH(_T)_</li> </ul>
Corresponding tag FB	<ul style="list-style-type: none"> <li>• M+M_PID(_T)</li> <li>• M+M_PID_DUTY(_T)</li> <li>• M+M_2PID(_T)</li> <li>• M+M_2PID_DUTY(_T)</li> <li>• M+M_2PIDH(_T)_</li> </ul>	<ul style="list-style-type: none"> <li>• M+M_2PIDH(_T)_</li> </ul>



For the execution and status check of the auto tuning, refer to the following.

PX Developer Version 1 Operating Manual (Monitor Tool)

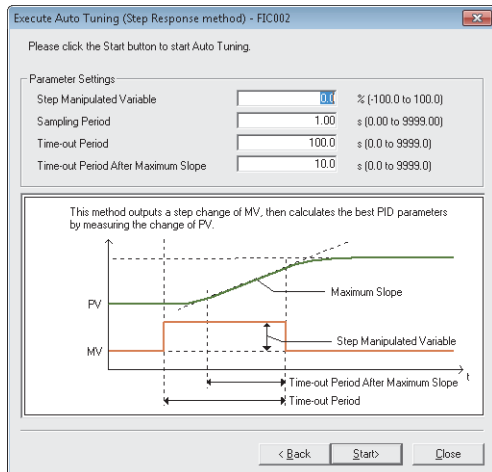
## Step response method

### ■ Operation method and processing details

Perform the following procedure after the manipulated value is held and the process variable is stabilized.

1. Display the auto tuning window of the PX Developer monitor tool.

☞ [Control Panel] ⇒ [Details] button on the faceplate ⇒ [Auto Tuning] button ⇒ [Select Auto Tuning Operations] window ⇒ Select [Executes Auto Tuning by Step Response method]. ⇒ [Next] button



2. Set the following items and click the [Start] button.

- Step manipulated variable (AT1STEPMV)
- Sampling period (AT1ST) (seconds)  
(PV data collection period during tuning)
- Time-out period (AT1TOUT1) (seconds)
- Time-out period after maximum slope (AT1TOUT2) (seconds)

3. The step manipulated variable is output from the current manipulated value in step form.

4. The manipulated value returns to its original value automatically after auto tuning is completed. P, I, and D constants generated from auto tuning are automatically saved in the tag data.

A

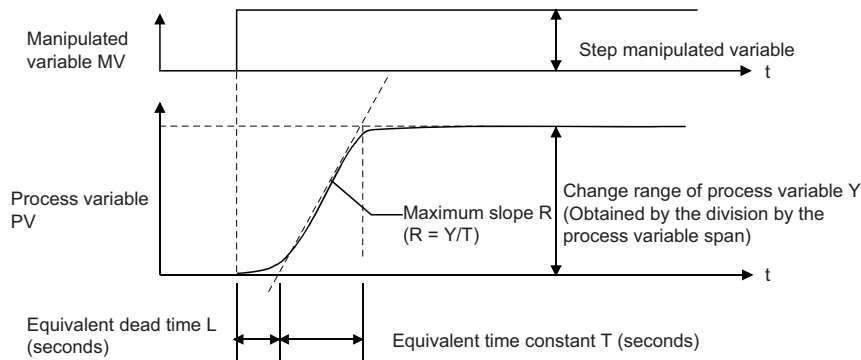
### Point

- PID constants are automatically overwritten after auto tuning. Save the previous PID constants in advance as necessary.
- Auto tuning stops automatically when an alarm occurs.

## ■ Operation details of step response method

The manipulated value is output in step form to the actual plant, and each constant is determined according to the maximum slope and equivalent dead time.

The following operations are automatically performed. The generated value is saved in P, I, and D areas of the tag data.



- Constants calculation with ZN method (Ziegler-Nichols' step response method)

The P/PI/PID control type is determined by I and D values of the tag data before auto tuning.

Condition		Control type	Constant		
Integral time (I)	Derivative time (D)		Proportional gain (P)	Integral time (I) (seconds)	Derivative time (D) (seconds)
I = 0 (Integral time ∞)	D = 0	P control	$\frac{1.0}{R \times L} \times \frac{ \text{Step manipulated variable} }{100}$	0 (Integral time ∞)	0
I > 0	D = 0	PI control	$\frac{0.9}{R \times L} \times \frac{ \text{Step manipulated variable} }{100}$	3.33L	0
I > 0	D > 0	PID control	$\frac{1.2}{R \times L} \times \frac{ \text{Step manipulated variable} }{100}$	2L	0.5L

### Ex.

Auto tuning is executed when the tag data before auto tuning are I > 0 and D = 0.

(For I > 0 and D = 0, the PI control calculates constants.)

Item	Value
Step manipulated variable	20%
Equivalent dead time L	8 seconds
Equivalent time constant T	16 seconds
The variation width of process variable Y	0.25
Maximum slope R	$0.25 / 16 = 0.016$

$$\text{Proportional gain (P)} = \frac{0.9}{R \times L} \times \frac{|\text{Step manipulated variable}|}{100} = \frac{0.9}{0.016 \times 8} \times \frac{20}{100} = 1.4$$

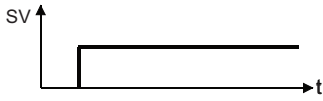
Integral time (I) = 3.33L = 3.33 × 8 = 26.6 seconds, derivative time (D) = 0 second



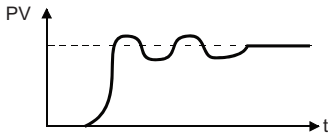
## ■ Fine tuning after auto tuning

After auto tuning is completed, the change of process variable (PV) in relation to the setting value (SV) is observed on the tuning setting execution window of the PX Developer monitor tool. The optimal values are obtained by adjusting the P, I, and D values.

The response of the process variable (PV) corresponding to the change of the setting value (SV) is observed.

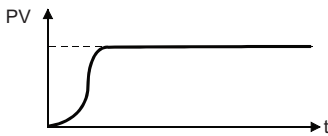


- Response is quick, but oscillatory.

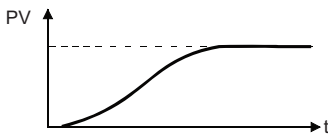


Fine tuning when the response is quick, but oscillatory

- Proportional gain: Reduce (Proportional effect is reduced.)
- Integral time: Increase (Integral effect is reduced.)
- Optimal value



- Response is slow



Fine tuning when the response is slow

- Proportional gain: Increase (Proportional effect is increased.)
- Integral time: Reduce (Integral effect is increased.)

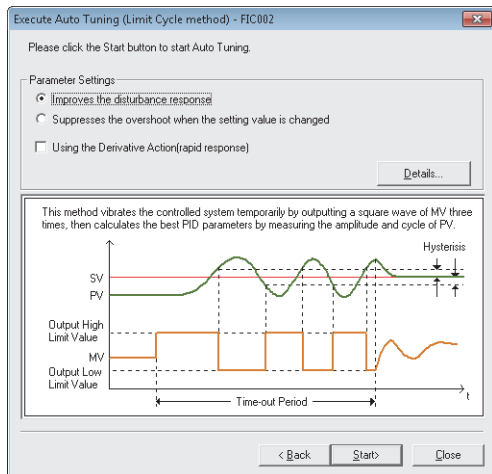
When the derivative action is applied, the derivative time is adjusted with the stability and rapid response check. (Increasing the derivative time increases the derivative effect, and reducing the derivative time reduces the derivative effect.)

## Limit cycle method

### ■ Operation method and processing details

1. Display the [Execute Auto Tuning (Limit Cycle method)] window of the PX Developer monitor tool.

☞ [Control Panel] ⇒ [Details] button on the faceplate ⇒ [Auto Tuning] button ⇒ [Select Auto Tuning Operations] window ⇒ Select [Executes Auto Tuning by Limit Cycle method]. ⇒ [Next] button



2. Set the following items and start auto tuning.

The control type is determined by selecting either of "Improves the disturbance response" or "Suppresses the overshoot when the setting value is changed", and whether to check "Using the Derivative Action (rapid response)".

○: Selected, ×: Not selected

Control type	Improves the disturbance response	Suppresses the overshoot when the setting value is changed	Using the Derivative Action (rapid response)
Constant-value PI control	○	×	×
Constant-value PID control	○	×	○
Follow-up PI control	×	○	×
Follow-up PID control	×	○	○

Set the following items on the detail setting window of the limit cycle method which is displayed by clicking the [Details] button.

- Output high limit (AT2MVH)
- Output low limit (AT2MVL)
- Hysteresis (AT2HS)
- Time-out period (AT1TOUT1) (seconds)

3. The high and low limit values of the manipulated value are repeatedly output.

Even if the value set to AT2MVH/AT2MVL exceeds MH/ML, the manipulated value is output within the range of MH to ML.

4. The manipulated value returns to its original value automatically after auto tuning is completed.

Values for proportional gain (Kp), integral time (Ti), and derivative time (Td) which are calculated by auto tuning are set automatically.

#### Point

- PID constants are automatically overwritten after auto tuning.
- Auto tuning stops automatically when an alarm occurs.
- MV output values return to the values at start when auto tuning is completed or interrupted.

## ■ Operation details of the limit cycle method

### ● Generation and measurement of the limit cycle waveform

In the AUTO TUNING mode, the PV limit cycle waveform is generated by performing the 2-position ON/OFF operation of MV output three times.

The 2-position ON/OFF operation is performed under the following conditions.

Control operation	First MV output	2-position ON/OFF operation	Remarks
Reverse action (PN = 0)	<ul style="list-style-type: none"> <li>■ <math>PV \leq SV</math> MV = Output high limit (AT2MVH)</li> <li>■ <math>PV &gt; SV</math> MV = Output low limit (AT2MVL)</li> </ul>	<ul style="list-style-type: none"> <li>■ <math>PV \geq SV + \text{Hysteresis (AT2HS)}</math> MV = Output low limit (AT2MVL)</li> <li>■ <math>PV \leq SV - \text{Hysteresis (AT2HS)}</math> MV = Output high limit (AT2MVH)</li> </ul>	For operation images, refer to the following examples.
Direct action (PN = 1)	<ul style="list-style-type: none"> <li>■ <math>PV \leq SV</math> MV = Output low limit (AT2MVL)</li> <li>■ <math>PV &gt; SV</math> MV = Output high limit (AT2MVH)</li> </ul>	<ul style="list-style-type: none"> <li>■ <math>PV \geq SV + \text{Hysteresis (AT2HS)}</math> MV = Output high limit (AT2MVH)</li> <li>■ <math>PV \leq SV - \text{Hysteresis (AT2HS)}</math> MV = Output low limit (AT2MVL)</li> </ul>	The MV high limit and low limit values at PN = 0 are reversely output.

The PV oscillation waveform data for the limit cycle method at the first 2-position ON/OFF operation is ignored. The oscillation amplitude  $X_c$  and oscillation period  $T_c$  are measured by using the PV oscillation waveform data at the second and third 2-position ON/OFF operations.

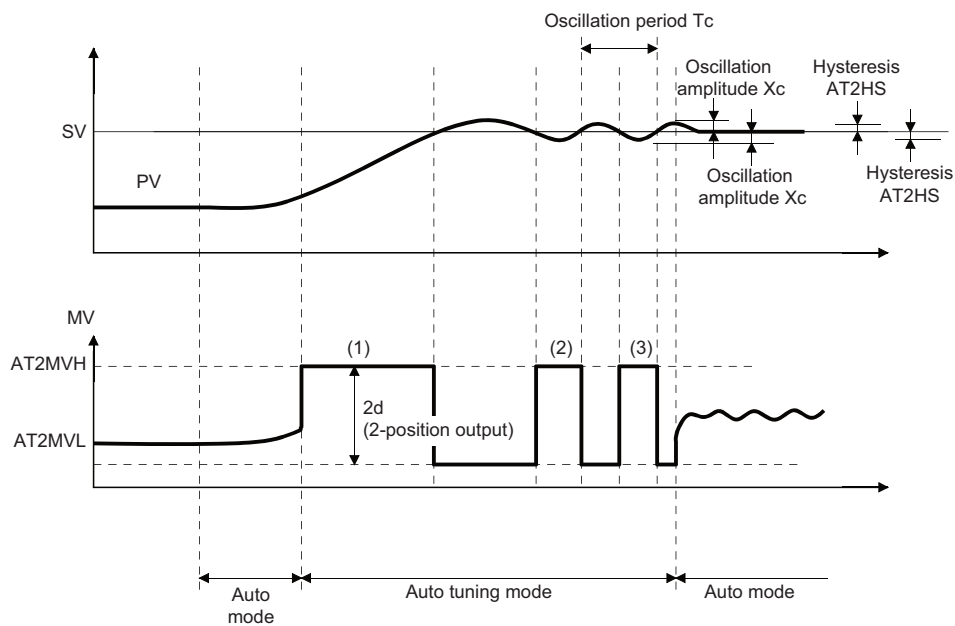
Auto tuning ends at the apex of the third PV oscillation waveform.

The setting value is calculated by using SVC (setting value (current)) at auto tuning start.

The hysteresis (AT2HS) functions as a minimum compensation value of the amplitude. Set it in advance according to the control target so that the oscillation period and oscillation amplitude are measured properly.

**Ex.**

$PV \leq SV$  (PN = 0)

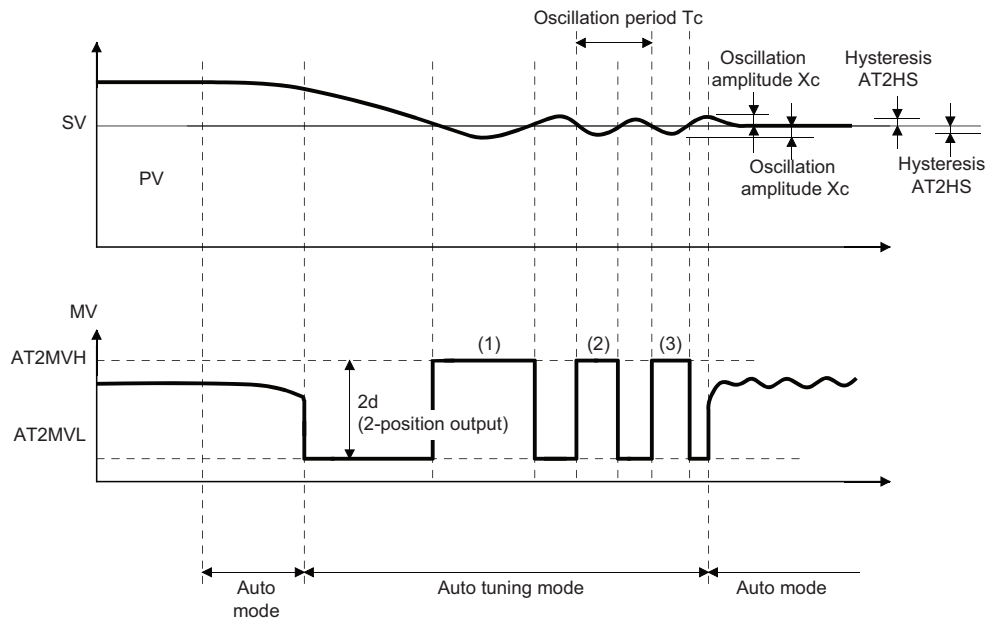


- (1) 1st 2-position operation
- (2) 2nd 2-position operation
- (3) 3rd 2-position operation

A

**Ex.**

PV > SV (PN = 0)



- (1) 1st 2-position operation
- (2) 2nd 2-position operation
- (3) 3rd 2-position operation

The oscillation amplitude  $X_c$  is calculated by measuring and averaging out plus side and minus side maximum values of  $|PV - SV|$ .

The output range  $d$  is calculated by  $(AT2MVH - AT2MVL) / 2$ .

● Calculation of threshold sensitivity and threshold period

Threshold sensitivity ( $K_u$ ) and threshold period ( $T_u$ ) are calculated from the measurement result of auto tuning by the limit cycle method.

$$K_u = 4d / (\pi \sqrt{X_c^2 - AT2HS^2})$$

$$T_u = T_c$$

● Calculation of optimal PID constant

The optimal PID constant is calculated from threshold sensitivity ( $K_u$ ) and threshold period ( $T_u$ ).

Values of proportional gain ( $K_p$ ), integral time ( $T_i$ ), and derivative time ( $T_d$ ) are calculated by using the coefficients specified for the control type (ATTTYPE) shown below.

Control type	Control operation	Control type (ATTTYPE)	Proportional gain ( $K_p$ )	Integral time ( $T_i$ )	Derivative time ( $T_d$ )	Empirical rule
Constant-value control	PI	1	$0.45K_u$	$0.83T_u$	0	Ziegler-Nichols' method
	PID	2	$0.6K_u$	$0.5T_u$	$0.125T_u$	
Follow-up control	PI	3	$0.3K_u$	$1.0T_u$	0	CHR method
	PID	4	$0.45K_u$	$0.6T_u$	$0.1T_u$	

$K_u$ : Process threshold sensitivity,  $T_u$ : Process threshold period

■ Fine tuning after auto tuning

Fine tuning of PID constants is the same as that for the step response method. (👉 Page 919 Fine tuning after auto tuning)

# Various controls

## Cascade control

The cascade control consists of the primary loop and secondary loop. This control detects disturbance entering secondary loop in an early stage and absorbs them into secondary loop to remove the effect on the process and improve the whole control performance.

Generally, the response speed of the secondary loop should be three times or more than that of the primary loop.

### Point

When tracking is required, connect CASOUT\_T of the primary loop with CASIN\_T of the secondary loop. In addition, settings for the following operation constants of the secondary loop are necessary.

- PID\_TRK
- PID\_SVPTN\_BO
- PID\_SVPTN\_B1

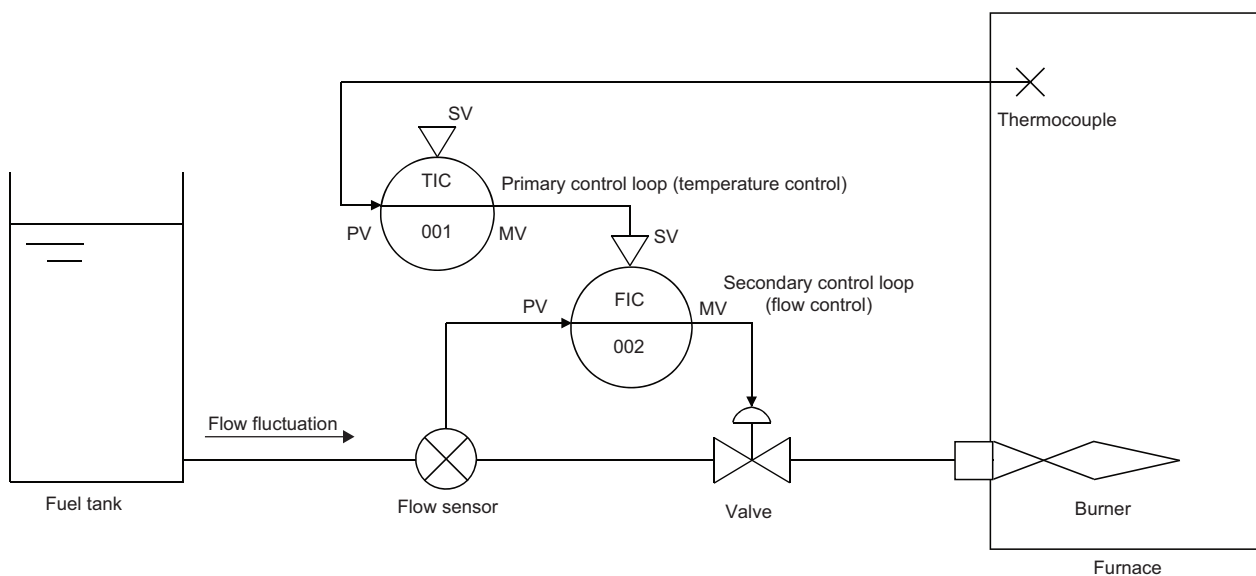
For details, refer to the following.

☞ Page 926 Cascade connection when tracking is required

### Ex.

When controlling the furnace temperature

The fuel supply variation is absorbed by flow control of the secondary control loop, and the response characteristics of the whole temperature control are improved.

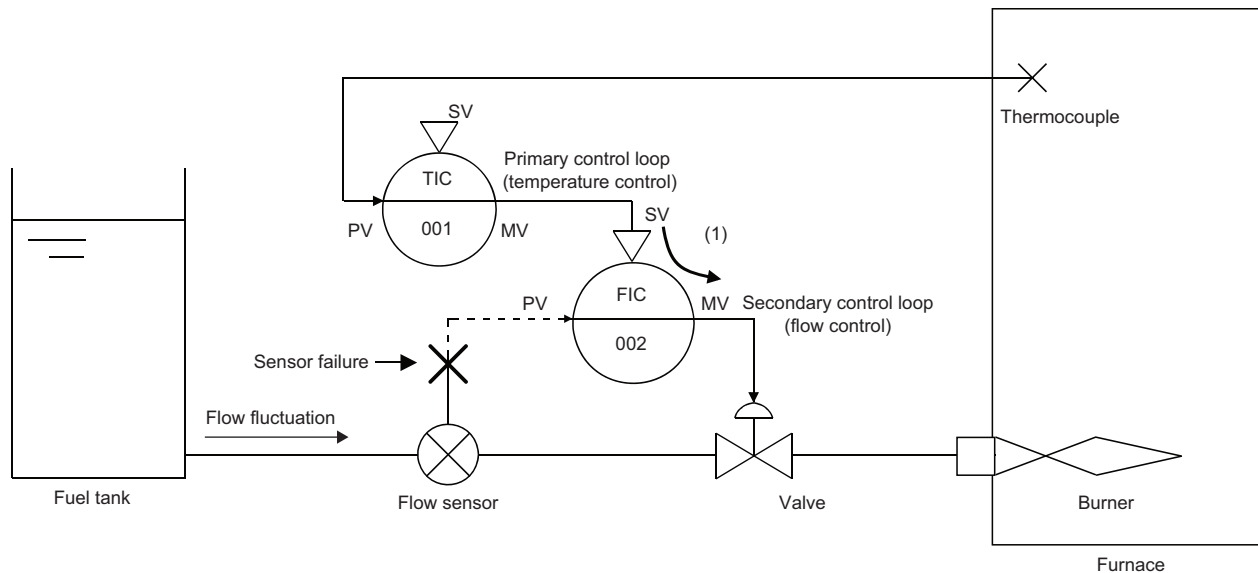


## ■ Cascade direct

For 2PIDH tags, the CASCADE DIRECT mode can be selected as the control mode. In the cascade direct control, the output value of the primary loop is directly output as that of the secondary loop in the cascade connection.

In case of the input sensor failure in the secondary loop, the output result of the primary loop is substituted for and directly output as the output value of the secondary loop since the PID operation result of the secondary loop will be improper.

The CASCADE DIRECT mode can be set with the tag of the secondary loop.

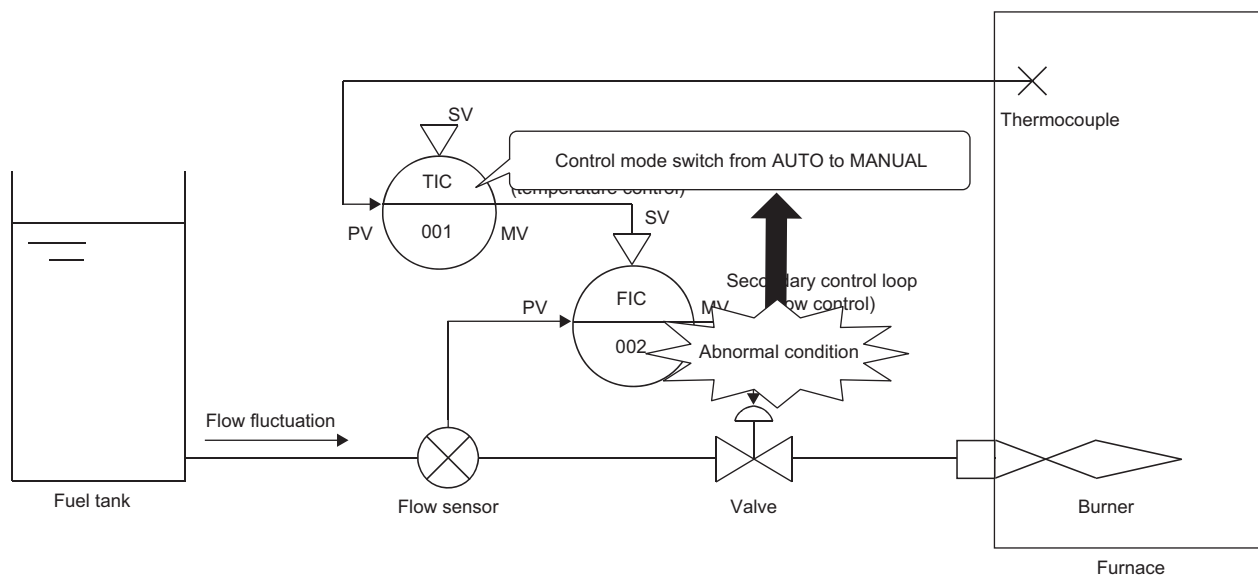


(1) MV of the primary loop is used as MV of the secondary loop.

## ■ Primary loop control mode switching

If an error (sensor error, output open, stop alarm) occurs in the secondary loop, the primary loop control mode can be forcibly switched to MANUAL.

Further, when the I/O mode is changed to tag stop, the primary loop control mode can be switched to MANUAL even when the control mode has been changed.



Set the following for the operation constants of the secondary loop tag FB.

Operation constant	Description	Setting value
MCHGPRMRY_SEA_EN	When a sensor error occurs, switches the primary loop to MANUAL mode	TRUE: Switch FALSE: Do not switch
MCHGPRMRY_OOA_EN	When output open occurs, switches the primary loop to MANUAL mode	
MCHGPRMRY_SPA_EN	When a stop alarm occurs, switches the primary loop to MANUAL mode	
MCHGPRMRY_TSTP_EN	When tag stop occurs, switches the primary loop to MANUAL mode	
MCHGPRMRY_CASCASDR_EN	When the control mode is changed (CAS/CASDR→AUT/MAN/CSV/CMV), switches the primary loop to MANUAL mode	

## Tracking

For tracking, there are two functions as shown below.

Item	Description
Bumpless function	At AUTO ↔ MANUAL mode switching, this function prevents the step change caused by the sharp change of the manipulated variable (MV) output and allows MV bumpless switching.
Output limiter processing function	This function sets the upper/lower limit for the manipulated variable (MV) that is output by PID operation in the AUTO mode. The output limiter processing function is valid only in the AUTO mode. It cannot be executed in the MANUAL mode. Even if the primary loop is in the AUTO mode, the output limiter processing function is not executed while the primary loop is being tracked by the secondary loop since the tracking data is stored as the manipulated value.

### Point

When tracking is required, connect CASOUT\_T of the primary loop with CASIN\_T of the secondary loop. In addition, settings for the following operation constants of the secondary loop are necessary.

- PID\_TRK
- PID\_SVPTN\_BO
- PID\_SVPTN\_B1

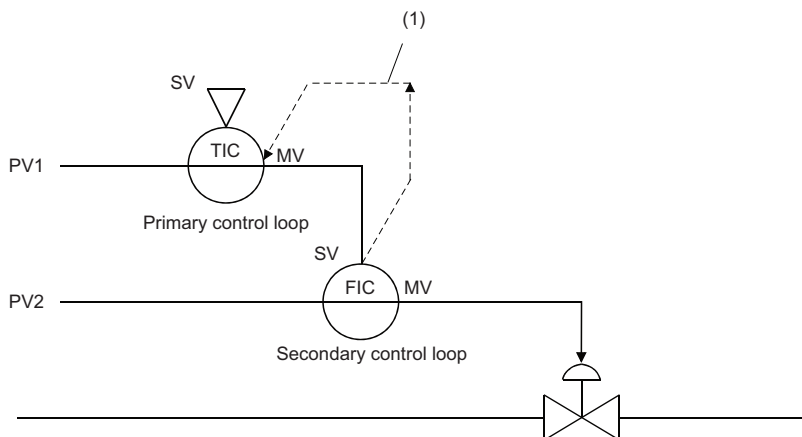
For details, refer to the following.

☞ Page 926 Cascade connection when tracking is required

### Ex.

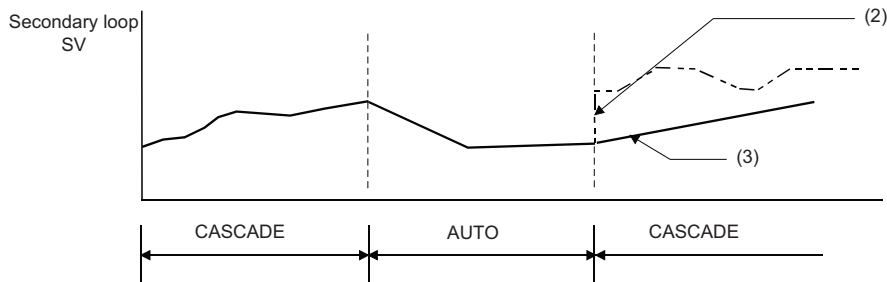
#### Tracking of cascade loop

In the control loops constituting the cascade loop, SV of the secondary loop is transferred to MV of the primary loop to prevent a sharp change of SV at the control mode switching in the secondary loop.



(1) If the control mode of the secondary control loop is other than CASCADE and tracking processing function is valid, SV of the secondary loop is transferred to MV of the primary loop, and tracking processing is performed. When the operation mode of the secondary loop is switched from AUTO to CASCADE, MV sharp change can be prevented since SV of the secondary loop is the same as MV of the primary loop.

The mode switching of the secondary loop is: CASCADE ↔ AUTO ↔ MANUAL. The MANUAL mode can be switched to the CASCADE mode through AUTO mode. The CASCADE mode can be switched to the MANUAL mode through AUTO mode.

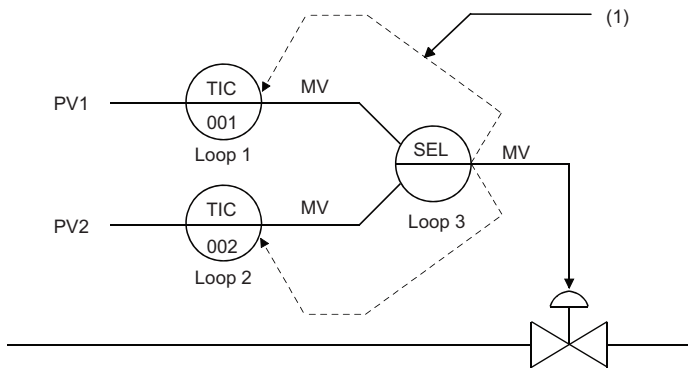


(2) Not tracked: SV of the secondary loop is sharply changed.

(3) Tracked: SV of the secondary loop is not sharply changed.

**Ex.**

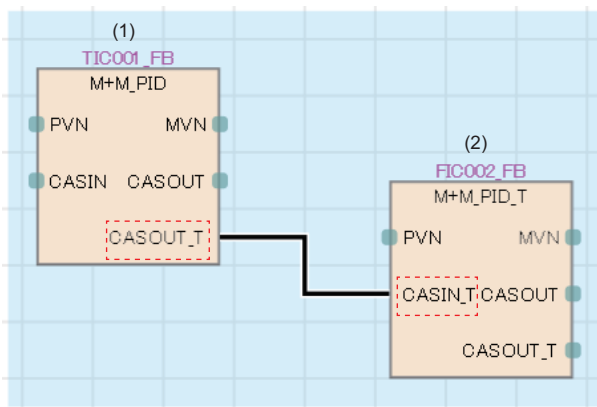
Tracking of loop selector



(1) MV of SEL (loop selector) is transferred to the primary loop when the control mode of SEL is manual (MANUAL, COMPUTER MV) or when the variation rate and high/low limiter alarm occurs in other than the MANUAL (AUTO, CASCADE, COMPUTER SV) mode. The primary loop with tracking setting to the SEL is tracked.

**■ Cascade connection when tracking is required**

Connect CASOUT\_T of the primary loop with CASIN\_T of the secondary loop. The tag FB of secondary loop should be the one with CASIN\_T (such as (M+M\_PID\_T, M+M\_2PID\_T)).



(1) Primary loop  
(2) Secondary loop

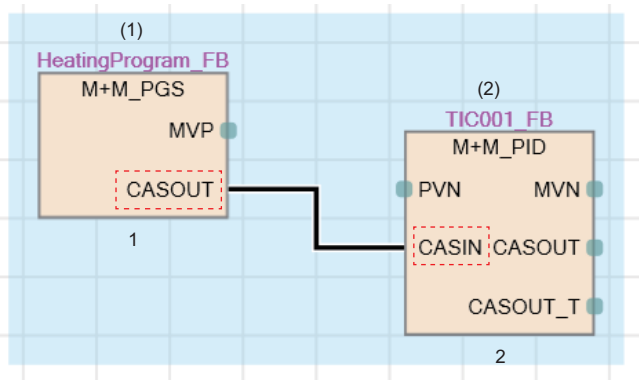
When tracking is executed, tracking setting for the operation constant of the secondary loop tag FB is required.

Variable name	Description	Range	Initial value	Setting for tracking
PID_TRK	Tracking flag	0: Without tracking 1: With tracking	0	1
PID_SVPTN_BO	Setting value (SV) used	TRUE: Not used FALSE: Used	TRUE	FALSE
PID_SVPTN_B1	Setting value (SV) pattern	TRUE: Not primary MV FALSE: Primary MV	TRUE	When the primary loop is the tag FB: FALSE (typically FALSE) When the primary loop is not the tag FB: TRUE



### ■ Cascade connection when tracking is not required

Connect CASOUT of the primary loop with CASIN of the secondary loop. The tag FB of the secondary loop should be the one with CASIN.



- (1) Primary loop
- (2) Secondary loop

When tracking is not executed, the operation constant of the secondary loop tag FB is as follows.

Variable name	Description	Range	Initial value	Setting for tracking
PID_SVPTN_BO	Setting value (SV) used	TRUE: Not used FALSE: Used	TRUE	FALSE

## I/O mode

The following describes the simulation function (SIMULATION mode), override function (OVERRIDE mode), and tag stop function (TAG STOP mode). For the overview of I/O mode, refer to the following.

☞ Page 28 I/O modes

### Simulation function (SIMULATION mode)

This function performs simulations without actual input/output for I/O modules.

Execute the simulation function (SIMULATION mode) after changing the mode to SIMULATION with I/O mode change of the faceplate. For details on I/O mode change, refer to the following.

📖 PX Developer Version 1 Operating Manual (Monitor Tool)

#### ■Function

Item	Description
For the loop tag FB	The loop control is executed by using MV output as feedback input while the PV external output and MV external input are not executed (with the input and output separated from the external). The loop test separated from the actual plant can be performed.
For the status tag FB	The input and output are separated from the external and the input signal is substituted a certain period after the output instruction is received. The actual valve ON/OFF instructions and response of answer signal can be simulated to confirm the control operation.

#### ■Creating simulation data

Item	Description
For the loop tag FB	The simulation data of the loop tag FB can be created by returning the simulation output (SIMOUT) to the simulation input (SIMIN) as variable in the tag FB.
For the status tag FB	Creation of program which returns input to output is not required. Set the simulation answer back signal period (SIMT) of tag data.

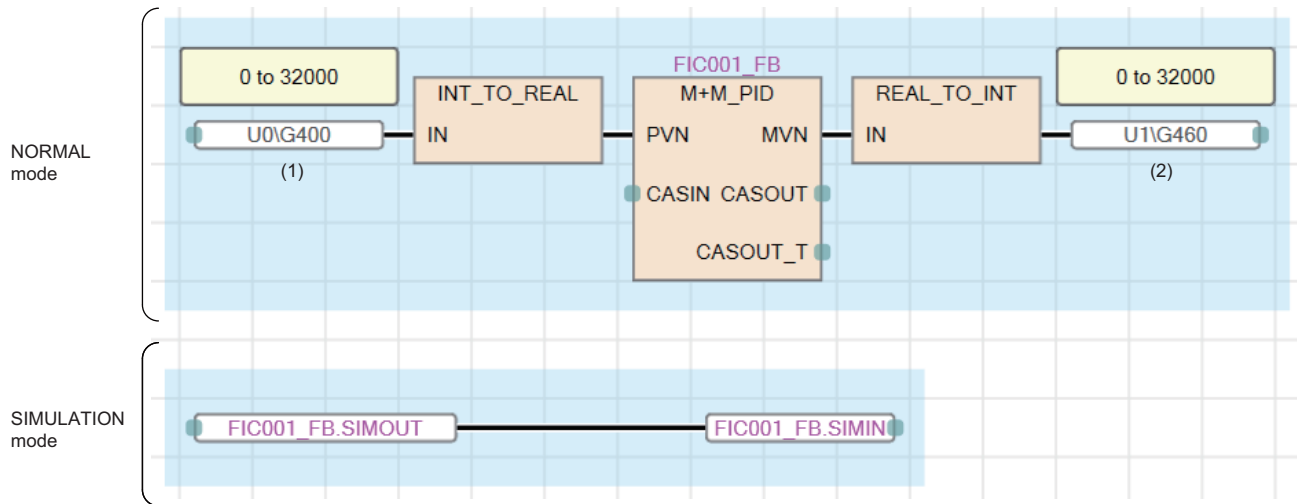
## ■ Sample programs of the loop tag FB simulation function (SIMULATION mode)

The following describes the sample programs for the simulation function (SIMULATION mode) when the loop tag FB is used.

**Ex.**

For the typical loop with the same range for PVN and MVN

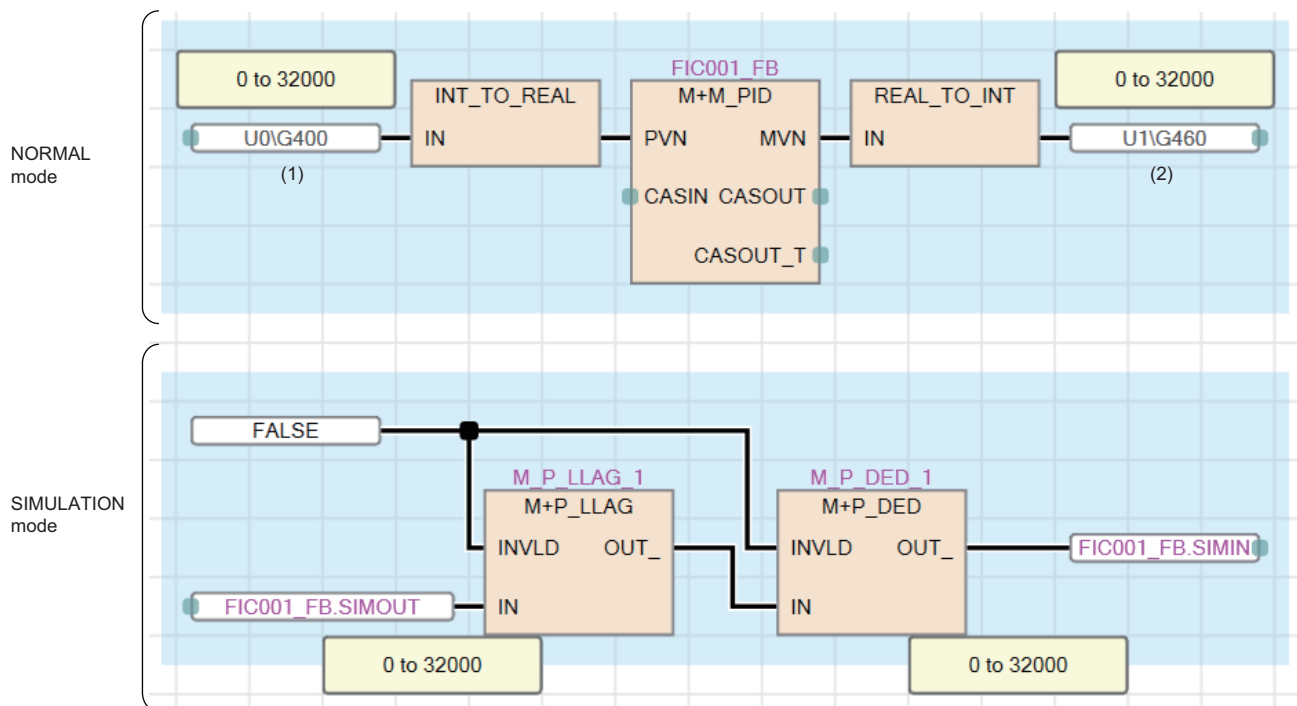
- Program example for the simulation function (SIMULATION mode)



(1) Analog input module CH1 digital output value

(2) Analog output module CH1 digital value

- Program example for the simulation function (SIMULATION mode) (with Primary delay + Dead time)



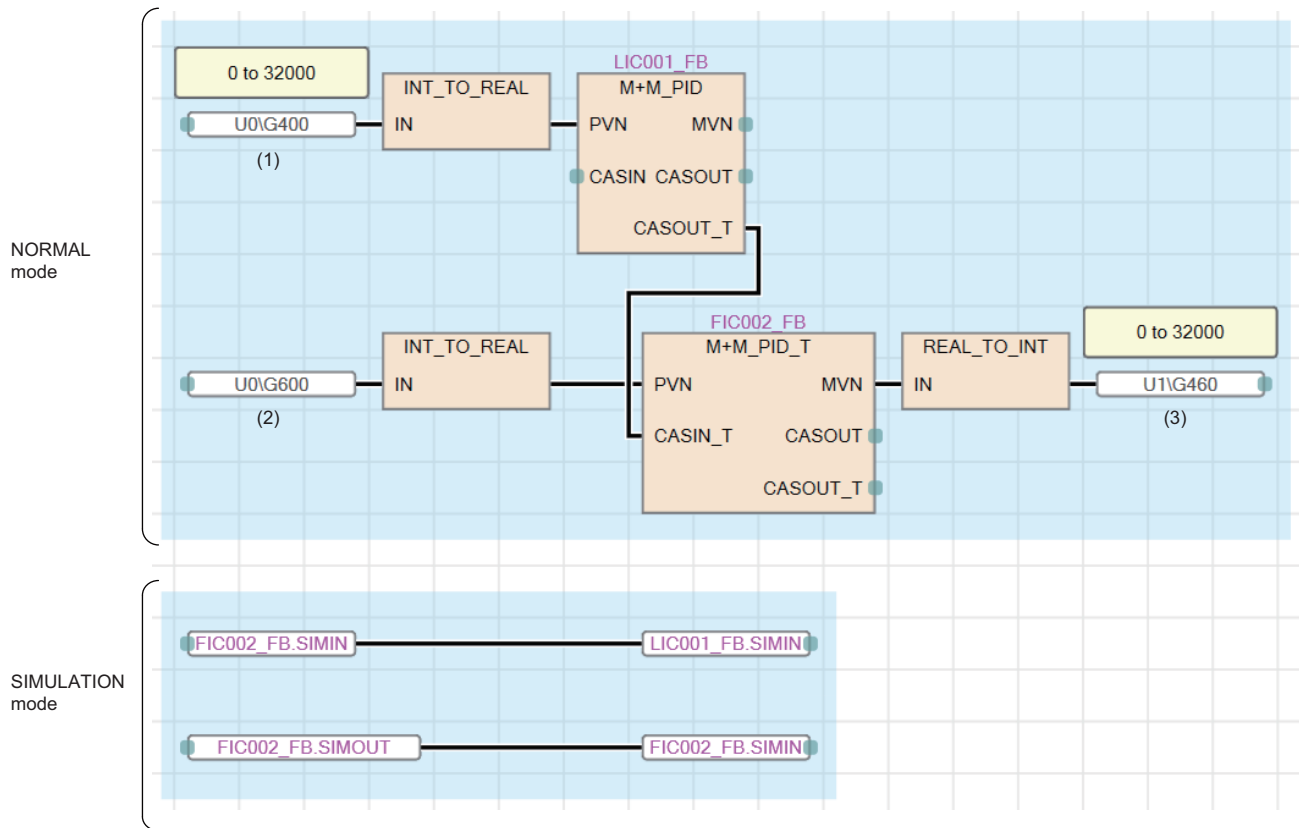
(1) Analog input module CH1 digital output value

(2) Analog output module CH1 digital value

A

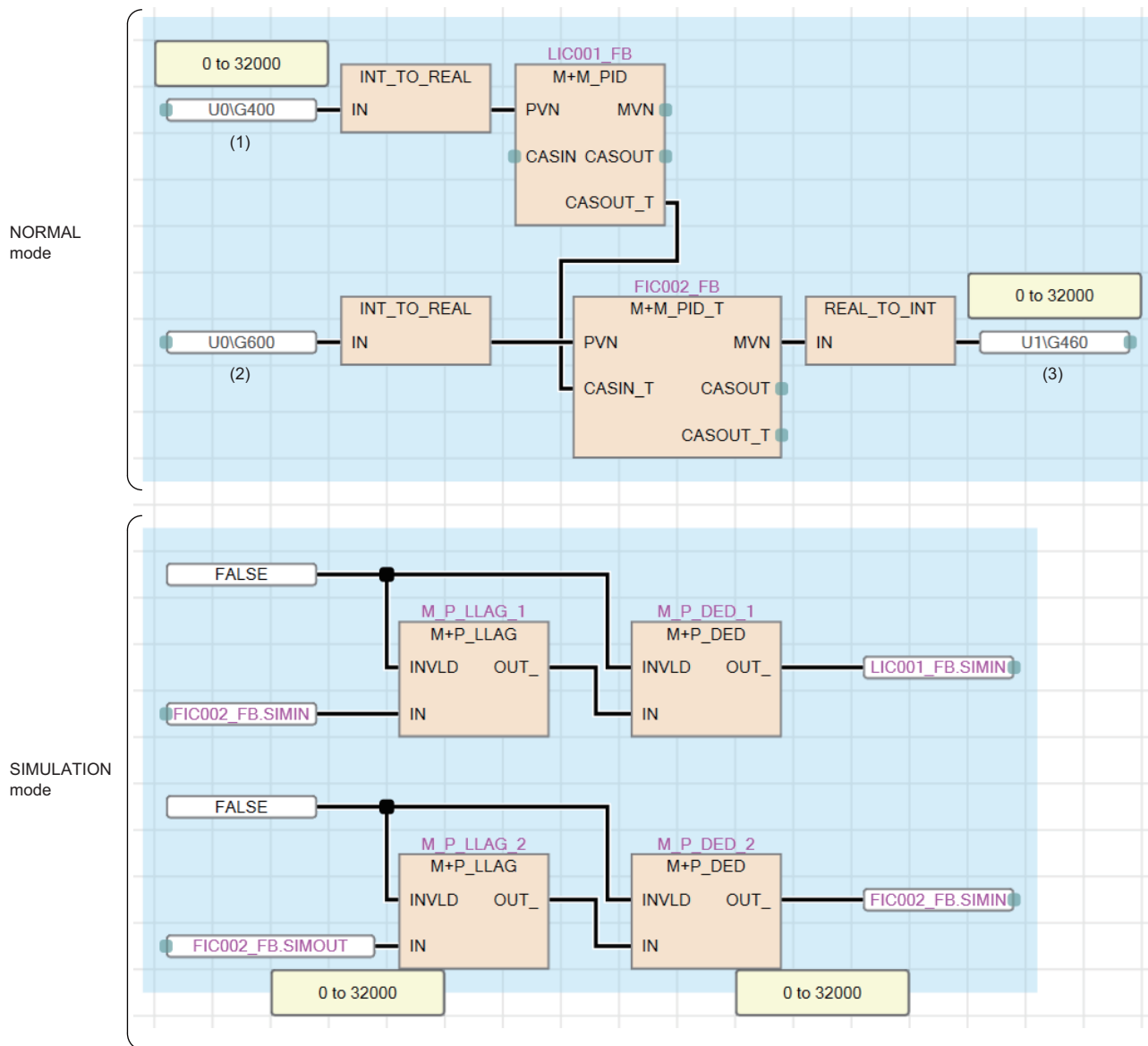
For the cascade connection

- Program example for the simulation function (SIMULATION mode)



- (1) Analog input module CH1 digital output value
- (2) Analog input module CH2 digital output value
- (3) Analog output module CH1 digital value

- Program example for the simulation function (SIMULATION mode) (with Primary delay + Dead time)



- (1) Analog input module CH1 digital output value
- (2) Analog input module CH2 digital output value
- (3) Analog output module CH1 digital value

**Point**

For the cascade connection, the simulation loopback input data of the secondary loop (FIC002.SIMIN in the example above) is used as the simulation input data of primary loop (LIC001.SIMIN in the example above).

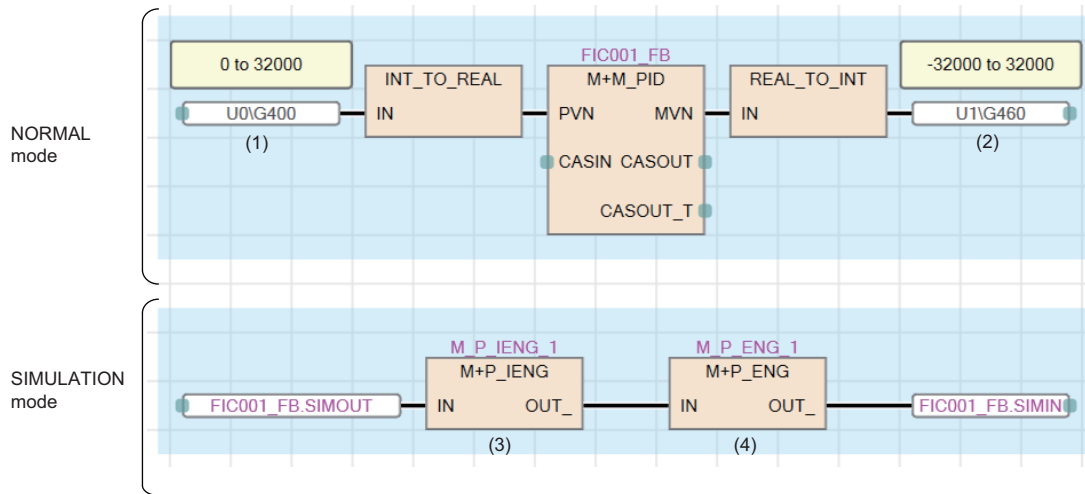


**Ex.**

When the ranges of PVN and MVN are different (the range of A/D conversion module (AIN\_4CH) is 0 to 32000, and the range of D/A conversion module (AOUT\_4CH) is -32000 to 32000)

- Program example for the simulation function (SIMULATION mode)

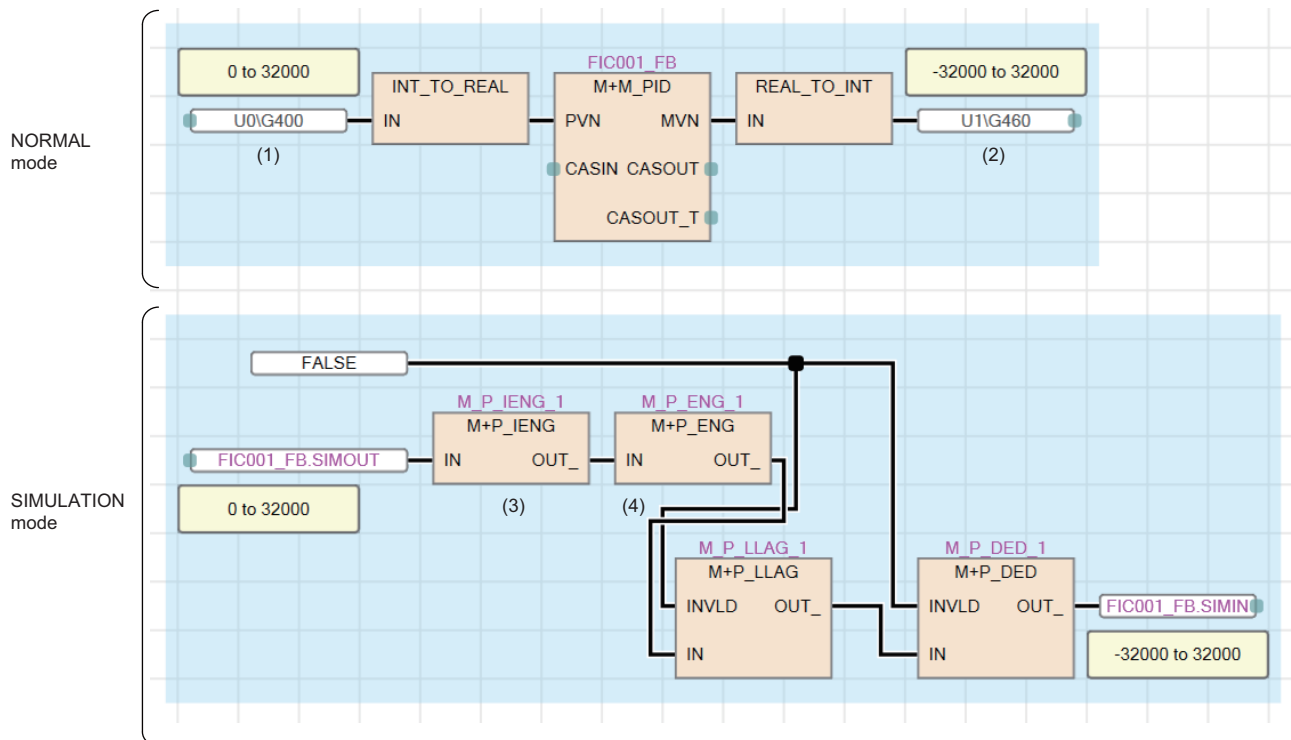
M+P\_ENG and M+P\_IENG convert FIC001\_FB.SIMOUT into FIC001\_FB.SIMIN for connection.



- (1) Analog input module CH1 digital output value
- (2) Analog output module CH1 digital value
- (3) The engineering value inverse conversion of 0 to 32000/0 to 100 (%)
- (4) The engineering value conversion of 0 to 100 (%) / -32000 to 32000

- Program example for the simulation function (SIMULATION mode) (with Primary delay + Dead time)

Convert the range from FIC001\_FB.SIMOUT to FIC001\_FB.SIMIN by M+P\_ENG and M+P\_IENG.



- (1) Analog input module CH1 digital output value
- (2) Analog output module CH1 digital value
- (3) The engineering value inverse conversion of 0 to 32000/0 to 100 (%)
- (4) The engineering value conversion of 0 to 100 (%) / -32000 to 32000

## Override function (OVERRIDE mode)

This function enables the PV setting on the pop-up tuning window of the PX Developer monitor tool when a correct input signal cannot be acquired due to a failure of the sensor, limit switch or A/D conversion module.

External outputs of the manual manipulated value and ON/OFF signal are performed.

Execute the override function after changing the mode to OVERRIDE with I/O mode change of the faceplate. For details on I/O mode change, refer to the following.

 PX Developer Version 1 Operating Manual (Monitor Tool)

### ■Function

Item	Description
For the loop tag FB	This function enables the PV setting on the pop-up tuning window of the PX Developer monitor tool when a correct PV input signal cannot be acquired due to a failure of the sensor or A/D conversion module. However, external output is performed. In this case, MV is output in the MANUAL mode. The function is used when the input signal is used for inter-lock conditions or transition conditions of batch sequence. For the PV setting, input the value to the tag monitor field on the pop-up tuning window of the PX Developer monitor tool.
For the status tag FB	This function enables to set input signals on the pop-up tuning window of the PX Developer monitor tool when a correct input signal cannot be acquired due to a poor contact of the valve open/close limit switch. However, external output is performed. The function is used when the input signal is used for inter-lock conditions or transition conditions of batch sequence.

### Point

For the loop tag FB, the override function can be operated in MANUAL mode only.  
Operations in other than the MANUAL mode cannot be operated.

## Tag stop function (TAG STOP mode)

This function stops input processing and loop control operation. Set it for the tag reserved for future use.

The tag stop function can be set by changing the I/O mode with the faceplate. For details on I/O mode change, refer to the following.

 PX Developer Version 1 Operating Manual (Monitor Tool)

### ■Function

The tag stop function can be used for 2PIDH, PGS2, SWM, PVAL, and HTCL tags.

This function is used for the tag which has been defined for future use with the engineering tool or has stopped.

For details, refer to the description of each function block.

# Stop alarm processing in loop control

## Stop alarm (SPA) overview

When an error related to the loop tag data such as a sensor error (SEA) occurs, the control mode can be changed to MANUAL forcibly by switching the stop alarm (SPA) from FALSE to TRUE.

Switch the stop alarm (SPA) from FALSE to TRUE using a user program as necessary.


When the stop alarm (SPA) is switched to TRUE, alarms which have occurred (MLA, MHA, DVLA, DPPA, PLA, PHA, LLA, HHA, SEA, and DMLA) will be automatically reset (TRUE → FALSE).

## Alarm (ALM) items of loop tag data

The following table lists alarm (ALM) items of loop tag data.

Offset	Label	Application	
+3	b0	MLA	Output Low Limit Alarm
	b1	MHA	Output High Limit Alarm
	b2	DVLA	Large Deviation Alarm
	b3	DPNA	Negative Variation Rate Alarm
	b4	DPPA	Positive Variation Rate Alarm
	b5	PLA	Input Low Limit Alarm
	b6	PHA	Input High Limit Alarm
	b7	LLA	Input Low Low Limit Alarm
	b8	HHA	Input High High Limit Alarm
	b9	SEA	Sensor Error Alarm
	bA	OOA	Output Open Alarm
	bB	DMLA	Output Variation Rate Limit Alarm
	bE	SPA	Stop Alarm

For the detailed tag data list, refer to the following.

 Page 812 Tag Data List

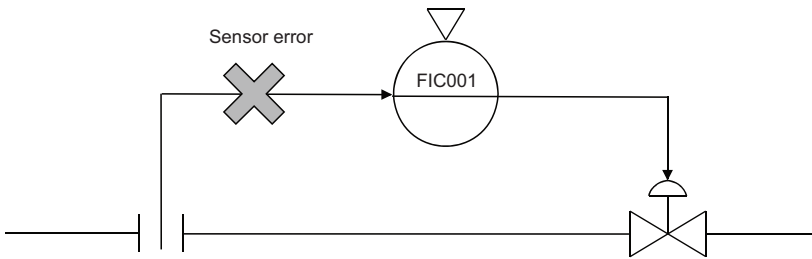
An alarm (ALM) consists of multiple Boolean variables. Boolean variables constituting an alarm (ALM) become global variables.



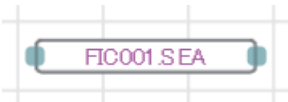
When using the stop alarm in a program, refer to the following example.

**Ex.**

When a loop sensor error (SEA) of FIC001 occurs



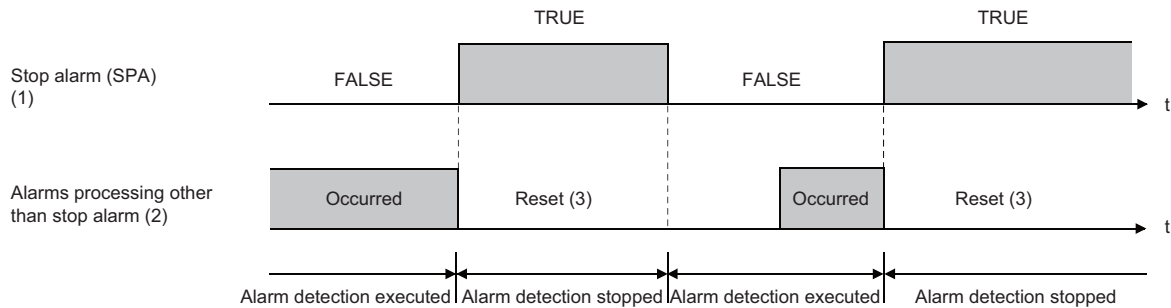
In the example above, the sensor error (SEA) can be acquired with the tag data label of the variable part.



## Relation between stop alarm (SPA) and other alarm processing

The following table shows the relation between the status of the stop alarm (SPA) and other loop alarm processing.

Stop alarm (SPA)	Result
TRUE	Alarms processing other than stop alarm (MLA, MHA, DVLA, DPPA, PLA, PHA, LLA, HHA, SEA, DMLA) are stopped, and then the alarm is automatically switched to FALSE.
FALSE	Alarm processing other than stop alarm (MLA, MHA, DVLA, DPPA, PLA, PHA, LLA, HHA, SEA, DMLA) is executed.



(1) Switch the stop alarm (SPA) to TRUE and FALSE using a program.

(2) The output open alarm (OOA) follows the processing in the program.

(3) When the stop alarm (SPA) is TRUE, the alarm detection processing is stopped. Alarms which have occurred are automatically reset.

**A**

# How to use the output open alarm

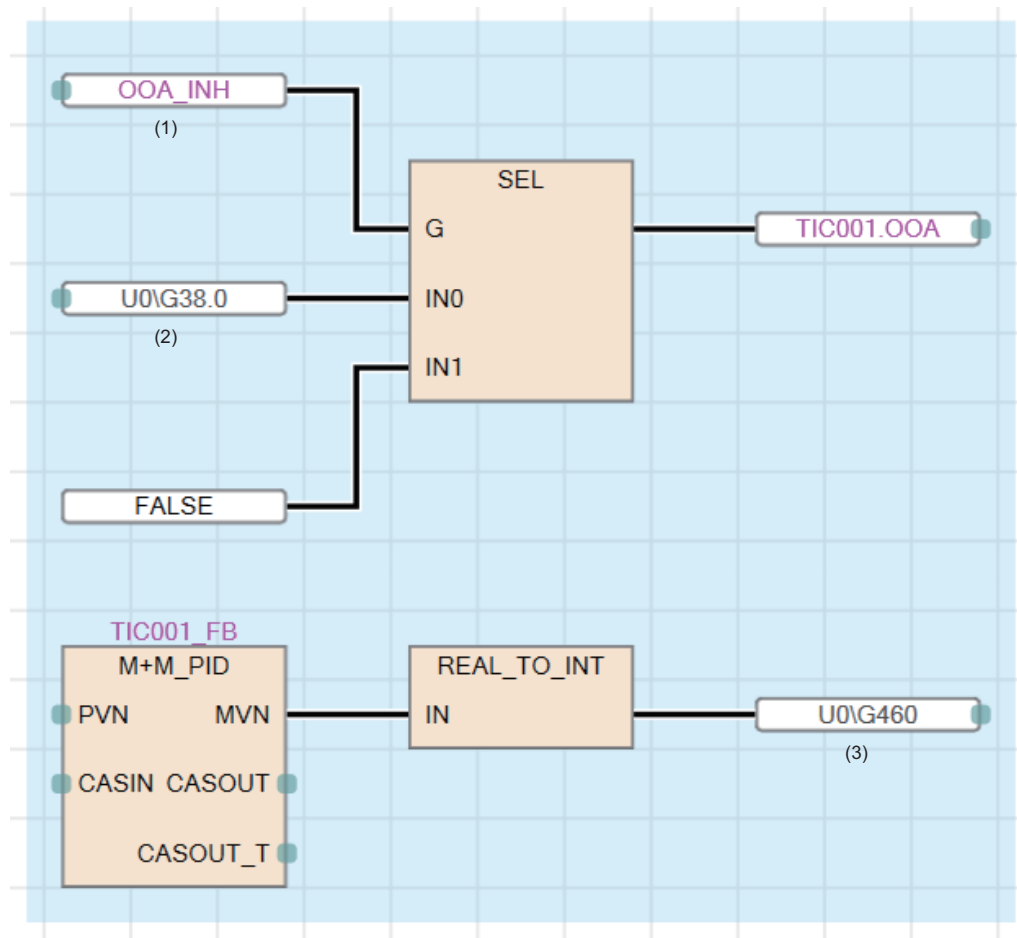
The output open alarm (OOA) of loop tag data is designed for displaying the wire break detection signal as an alarm on the loop tag FB of the output source when a disconnection is detected on the module FB on the output side.

The following shows a program example to feed back a signal from the module FB that is to detect a disconnection to the loop tag FB that is to display it as an alarm.

## Program example

Input the module access device on the output side (CH1 disconnection detection flag) to the tag item (OOA) of the loop tag FB.

For the output open alarm (OOA), the disable alarm detection flag does not exist. Therefore, create a program separately to enable or disable this alarm.



- (1) User-created variable
- (2) CH1 disconnection detection flag
- (3) CH1 digital value

# Program setter setting method

The following describes the setting method of the program setter (PGS) and multi-point program setter (PGS2).

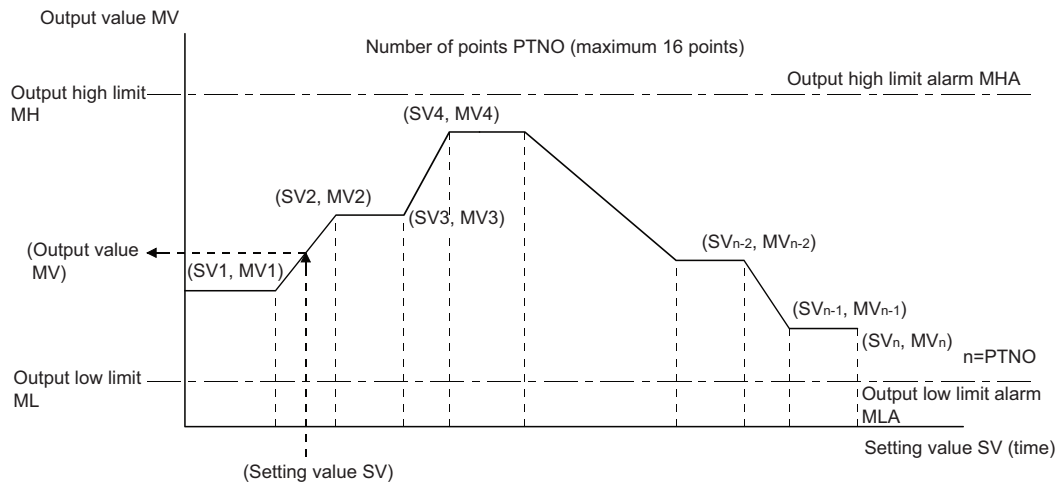
## Program setter (PGS)

### ■Operation method

The X-Y graph method is used. The output MV is calculated by using the X-Y graph function according to the SV (time) of the whole program.

### ■Relation between the program setting method and each variable

Register a program using the X-Y graph method shown below.



For details on each variable, refer to the following.

☞ Tag Data List

### ■Registration format

Register the break point data up to 16 points in real number (REAL).

### ■Time management

Set the time in units of seconds.

### ■Output high/low limit alarm

Place the output low limit alarm MLA and output high limit alarm MHA to bit 0 and bit 1 (standard locations) of the tag data +3 (ALM).

### ■Mode and operation type

Five control modes are available: MANUAL, COMPUTER MV, AUTO, CASCADE, and COMPUTER SV.

The operation type is CYCLIC in the CASCADE mode.

In the AUTO mode, HOLD or RETURN can be selected for the operation type.

## Multi-point program setter (PGS2)

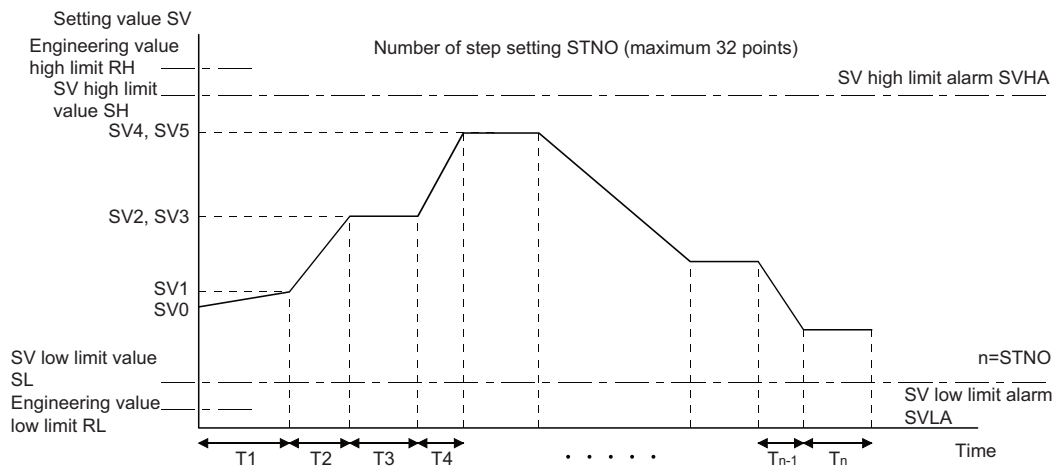
### ■ Operation method

Steps (time width and setting value) are registered and the progress of each step is managed.

The setting value (SV) is calculated according to the time in the step (T) in each step.

### ■ Relation between the program setting method and each variable

Register each step data as shown below. Set the program start point to SV0.



$T_n$ : Time in the step (T)

n: Executing step No. (STC)

The following main parameters are changed from the program setter (PGS).

- MV (Output value) → SV (Setting value)
- SV (Setting value) → STC (Executing step number) + T (Time in the step)

For details on each variable, refer to the following.

☞ Tag Data List

### ■ Registration format

Register the step data up to 32 points integer (INT).

The setting range is from -32768 to 32767.

### ■ Time management

Set the time in units of seconds or minutes. (Set at TUNIT in the tag data.)

### ■ Output high/low limit alarm

Place the SV low limit alarm SVLA and SV high limit alarm SVHA to bit 0 and bit 1 of the tag data +3 (ALM).

(Variable names differ from those for the program setter (PGS).)

### ■ Mode and operation type

Two control modes are available: MANUAL and AUTO.

In the AUTO mode, HOLD, RETURN, or CYCLIC can be selected for the operation type.

## Tight shut/full open function

To use the tight shut/full open function with a module that has the extended mode in the range setting, set the extended mode in the output range setting.

To use the tight shut/full open function with a module that does not have the extended mode in the range setting, set 0 to 20mA and 0 to 5V in the range setting and set the output conversion high/low limit values of the two-degree-of-freedom advanced PID control FB again.

### Point

When 4 to 20mA and 1 to 5V are regarded as a standard, only the tight shut (full open when MV reverse is valid) is applied since the upper limit side can output up to 100% (20mA, 5V) with output range setting of 0 to 20mA and 0 to 5V.

The following table shows a setting example for the two-degree-of-freedom advanced PID control FB when the signal of 4 to 20mA is regarded as a standard, tight shut output value is set to -20% (0.8mA).

For the analog module, set 0 to 20mA for the output range setting and 0 to 32000 for the resolution.

Item	Setting value	Remarks
Output Conversion High Limit (OUT3_NMAX)	32000.0	Equivalent to 20mA
Output Conversion Low Limit (OUT3_NMIN)	6400.0	Equivalent to 4mA
Tight Shut/Full Open Execution condition (OUT3_FOTS_EN)	TRUE	—
Output Value for Full Open (Unit: %) (OUT3_MVFO)	100.0	100%, since 20mA or more cannot be output.
Output Value for Tight Shut (Unit: %) (OUT3_MVTS)	-20.0	Equivalent to 0.8mA

# Appendix 4 Program Example

This section describes a program example with the process control function blocks.

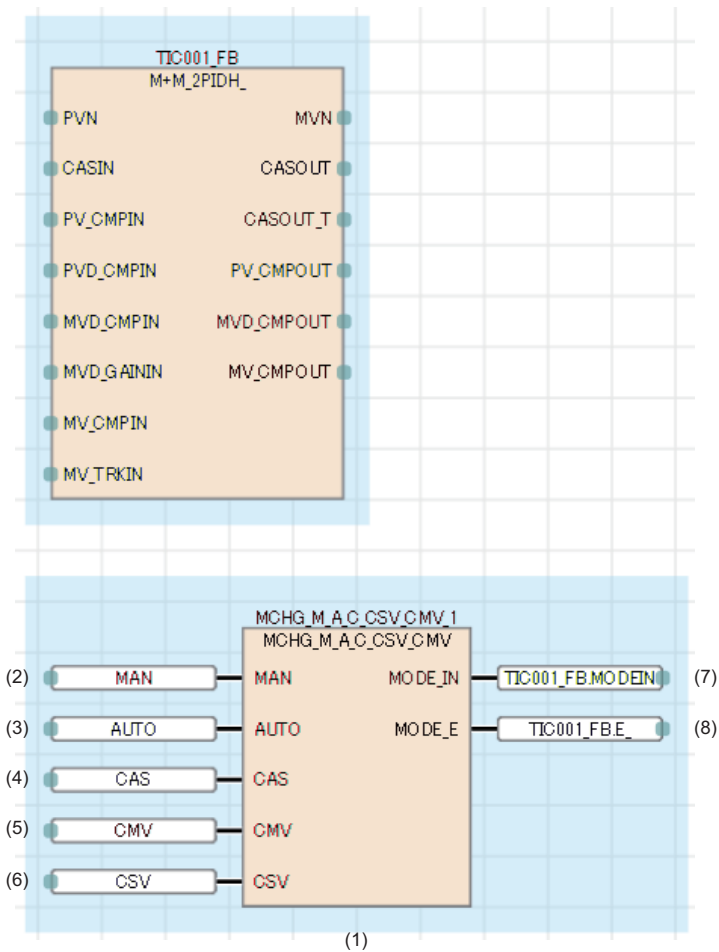


For details on the FBD/LD language, refer to the following.  
 MELSEC iQ-R Programming Manual (Program Design)

## Control mode switching

The following indicates the program which switches the control mode of two-degree-of-freedom advanced PID control (M+M\_2PIDH\_) to one of MANUAL, AUTO, CASCADE, COMPUTER MV, or COMPUTER SV in the control mode.

### Program example



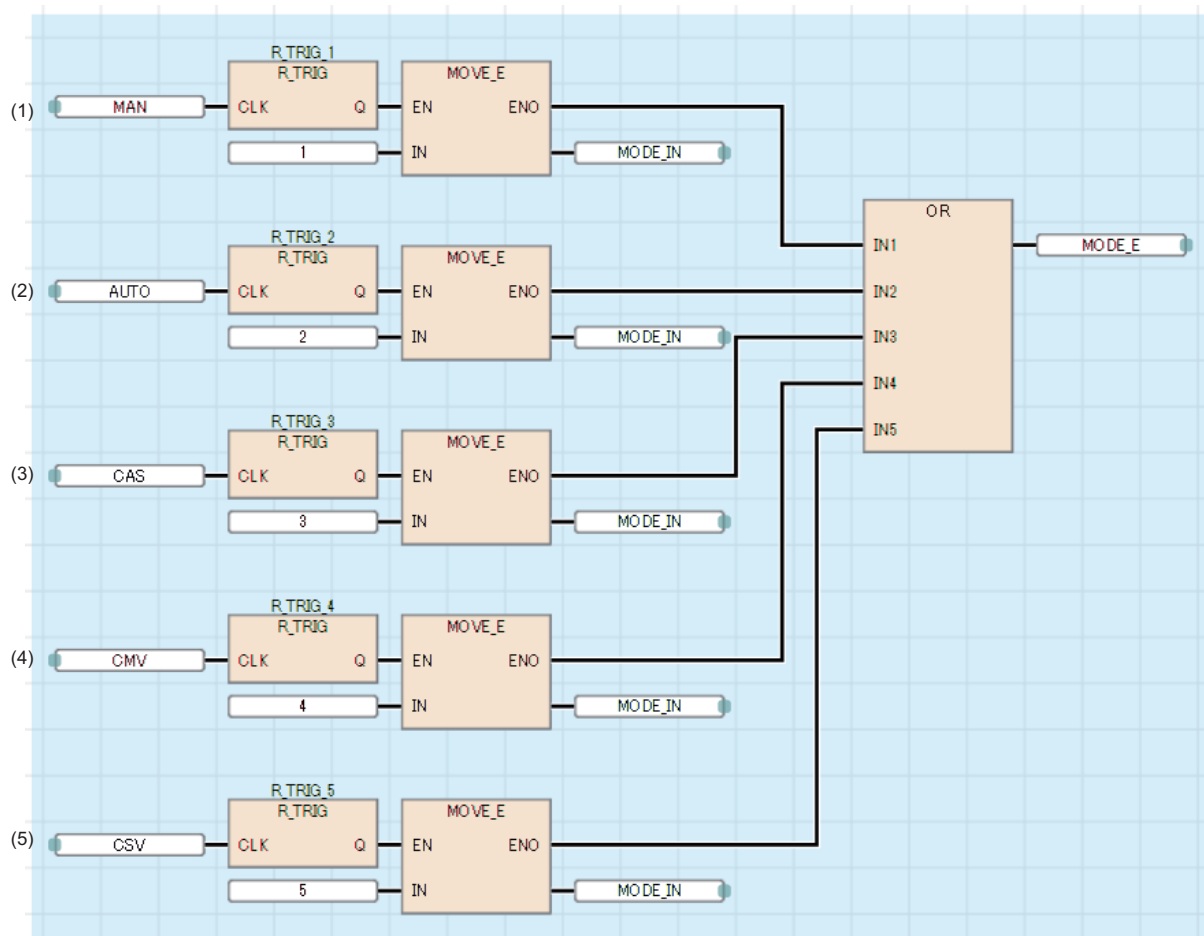
- (1) Function block (Page 941 Function block (MCHG\_M\_A\_C\_CSV\_CMV)) of control mode switching
- (2) MANUAL switching command (BOOL)
- (3) AUTO switching command (BOOL)
- (4) CASCADE switching command (BOOL)
- (5) COMPUTER MV switching command (BOOL)
- (6) COMPUTER SV switching command (BOOL)
- (7) Public variable: Mode change signal (INT)
- (8) Public variable: switching request (BOOL)

## Function block (MCHG\_M\_A\_C\_CSV\_CMV)

### Input/output variable

Variable name	Description	Type	Data type
MAN	TRUE: MANUAL	Input variable	BOOL
AUTO	TRUE: AUTO	Input variable	BOOL
CAS	TRUE: CASCADE	Input variable	BOOL
CMV	TRUE: COMPUTER MV	Input variable	BOOL
CSV	TRUE: COMPUTER SV	Input variable	BOOL
MODE_IN	Control mode (1: MANUAL, 2: AUTO, 3: CASCADE, 4: COMPUTER MV, or 5: COMPUTER SV)	Output variable	INT
MODE_E	Switching command (TRUE: execution)	Output variable	BOOL

### Program example



- (1) Switch the control mode to MANUAL when the signal for MAN changes from FALSE into TRUE.
- (2) Switch the control mode to AUTO when the signal for AUTO changes from FALSE into TRUE.
- (3) Switch the control mode to CASCADE when the signal for CAS changes from FALSE into TRUE.
- (4) Switch the control mode to COMPUTER MV when the signal for CMV changes from FALSE into TRUE.
- (5) Switch the control mode to COMPUTER SV when the signal for CSV changes from FALSE into TRUE.

#### Point

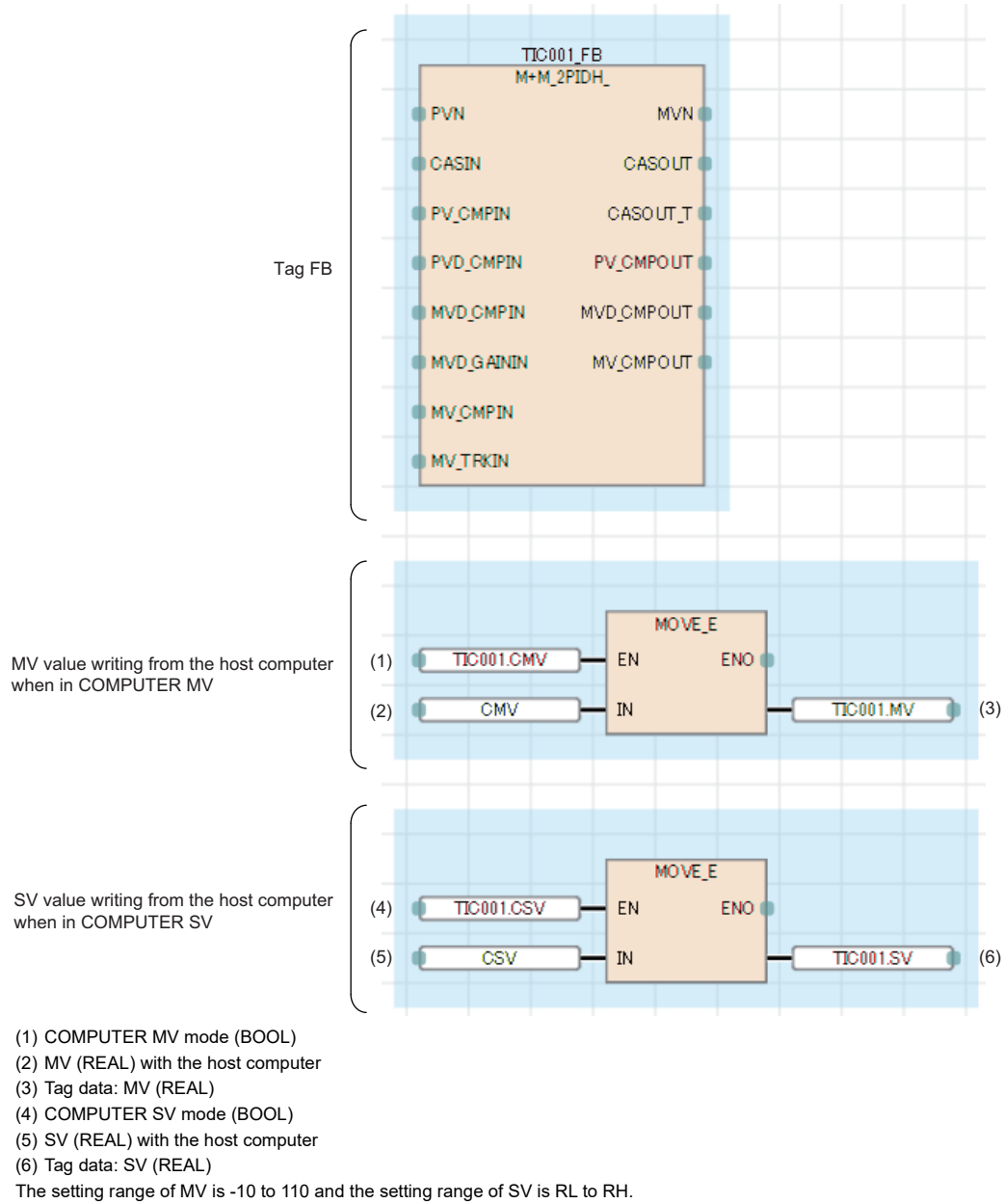
For the R\_TRIG (Rising edge detection), refer to the following.

MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

# Writing MV or SV with the host computer

The following indicates the program which writes MV or SV when the control mode of the tag FB is COMPUTER MV or COMPUTER SV.

## Program example

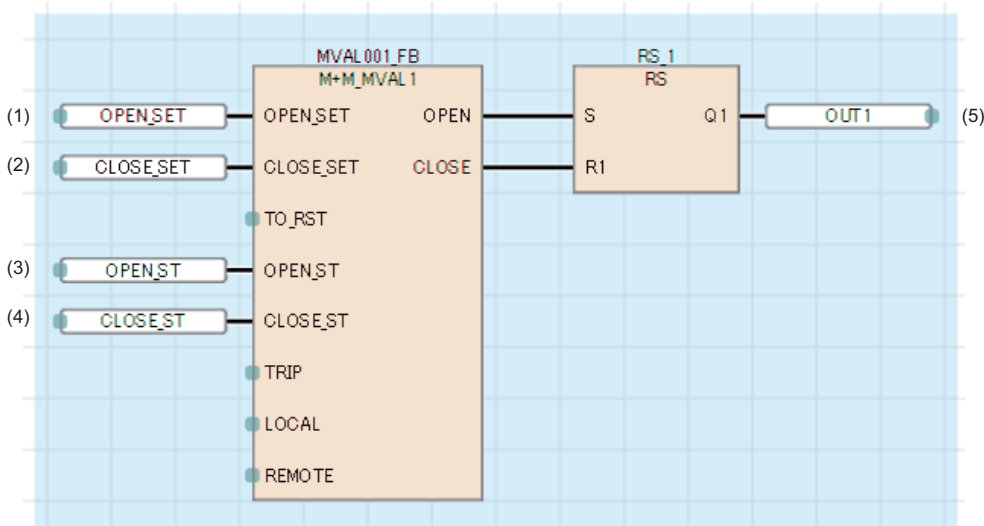




# Single solenoid

The following indicates the program which continuously outputs TRUE for open command and FALSE for close command when the control mode is AUTO.

## Program example



- (1) Open command (BOOL)
- (2) Close command (BOOL)
- (3) Open answer ( BOOL)
- (4) Close answer (BOOL)
- (5) Output (BOOL)  
Open: TRUE, Close: FALSE

### Point

For RS (Bistable function block (reset-dominant)), refer to the following.

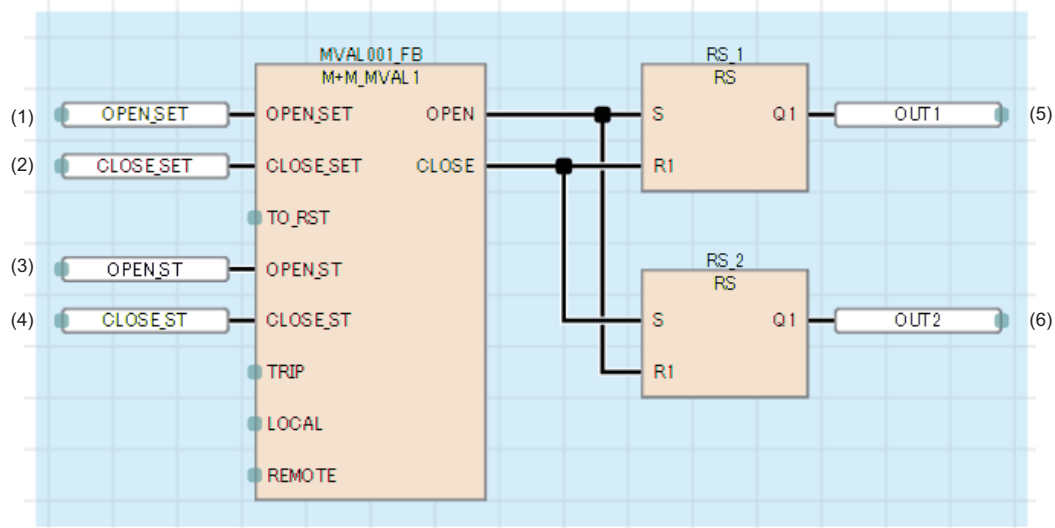
MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

A

# Double solenoid

The following indicates the program which continuously outputs TRUE into the open output (OUT1) for open command and TRUE into the close output (OUT2) for close command from outside when the control mode is AUTO.

## Program example



- (1) Open command (BOOL)
- (2) Close command (BOOL)
- (3) Open answer (BOOL)
- (4) Close answer (BOOL)
- (5) Output (BOOL)  
Open command: TRUE output continuously
- (6) Close output (BOOL)  
Close command: TRUE output continuously

**Point**

For RS (Bistable function block (reset-dominant)), refer to the following.

MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

# Appendix 5 Replacement of Other Format Projects (PX Developer-compatible Function Blocks)

When a PX Developer format project is used in GX Works3, the following instructions are automatically replaced by the compatible function blocks.\*1

\*1 These compatible function blocks are available on the Process CPU and the SIL2 Process CPU.  
GX Works3 with version "1.055H" or later and PX Developer with version "1.52E" or later, or GX Works3 with version "1.080J" or later and PX Developer with version "1.55H" or later support this feature.

**Point** For "GX Works3: 1.054G or earlier, PX Developer: 1.52E or later" or "GX Works3: 1.077F or earlier, PX Developer: 1.55H or later", because the instructions are replaced with program elements not supported by GX Works3 with the relevant version, an error occurs when a project is converted. Upgrade the versions of GX Works3 and PX Developer to the latest versions. (☞ Page 947 Program Example of Matching Operations with the Operations of PX Developer)

## Compatible function block

Function blocks used in FBD programs are replaced by the compatible function blocks.

PX Developer function block	Compatible function block	Program type		Reference
		ST	FBD	
EDGE_CHECK	EDGE_CHECK_PX	○	○	☞ PX Developer Version 1 Programming Manual
LATCH_BOOL	LATCH_BOOL_PX	○	○	
LATCH_REAL	LATCH_REAL_PX	○	○	
LATCH_WORD	LATCH_WORD_PX	○	○	
LATCH_DWORD	LATCH_DWORD_PX	○	○	
TOF_HIGH	TOF_HIGH_PX*1	○	○	
TOF_LOW	TOF_LOW_PX*1	○	○	
TON_HIGH	TON_HIGH_PX*1	○	○	
TON_LOW	TON_LOW_PX*1	○	○	
TP_HIGH	TP_HIGH_PX*1	○	○	
TP_LOW	TP_LOW_PX*1	○	○	

\*1 A part of operation of the compatible function block is different from that of the function block in PX Developer.



## ■ Operation difference

Compatible function block	Description
M+SR, M+RS	If the public variable IR is TRUE, after the start-up of the CPU module (powered off and on, reset), when an FB instance is called for the first time, this function block resets the output value in the output variable Q1. When the public variable IR is FALSE, this function block holds the output value in the output variable Q1.
TOF_HIGH_PX TOF_LOW_PX	The valid range for the OFF delay timer time <sup>*1</sup> depends on the long timer setting. Since the data type of the OFF delay timer time is TIME data type (32-bit value), the maximum value will fall within the range of TIME data type. <b>■ Minimum value</b> <ul style="list-style-type: none"> <li>Identical to the long timer setting value (ms). Note that if the long timer setting value is smaller than 1ms, the minimum value will be 1ms.</li> </ul> <b>■ Maximum value</b> <ul style="list-style-type: none"> <li>OFF delay timer time (ms) <math>\leq</math> 2147483647 (ms) <math>\times</math> Long timer setting value (ms)</li> </ul> If the OFF delay timer time exceeds the valid range, an operation error occurs. (Error code: 3401H) If an operation error occurs, correct the long timer setting value.
TON_HIGH_PX TON_LOW_PX	The valid range for the ON delay timer time <sup>*1</sup> depends on the long timer setting. Since the data type of the ON delay timer time is TIME data type (32-bit value), the maximum value will fall within the range of TIME data type. <b>■ Minimum value</b> <ul style="list-style-type: none"> <li>Identical to the long timer setting value (ms). Note that if the long timer setting value is smaller than 1ms, the minimum value will be 1ms.</li> </ul> <b>■ Maximum value</b> <ul style="list-style-type: none"> <li>ON delay timer value (ms) <math>\leq</math> 2147483647 (ms) <math>\times</math> Long timer setting value (ms)</li> </ul> If the ON delay timer time exceeds the valid range, an operation error occurs. (Error code: 3401H) If an operation error occurs, correct the long timer setting value.
TP_HIGH_PX TP_LOW_PX	The valid range for pulse width time <sup>*1</sup> depends on the long timer setting. Since the data type of the pulse width time is TIME data type (32-bit value), the maximum value will fall within the range of TIME data type. <b>■ Minimum value</b> <ul style="list-style-type: none"> <li>Identical to the long timer setting value (ms). Note that if the long timer setting value is smaller than 1ms, the minimum value will be 1ms.</li> </ul> <b>■ Maximum value</b> <ul style="list-style-type: none"> <li>Pulse width time (ms) <math>\leq</math> 2147483647 (ms) <math>\times</math> Long timer setting value (ms)</li> </ul> If the pulse width time exceeds the valid range, an operation error occurs. (Error code: 3401H) If an operation error occurs, correct the long timer setting value.

\*1 For the OFF delay timer time, ON delay timer time, and pulse width time, refer to the following.

 PX Developer Version 1 Programming Manual

# Appendix 6 Program Example of Matching Operations with the Operations of PX Developer

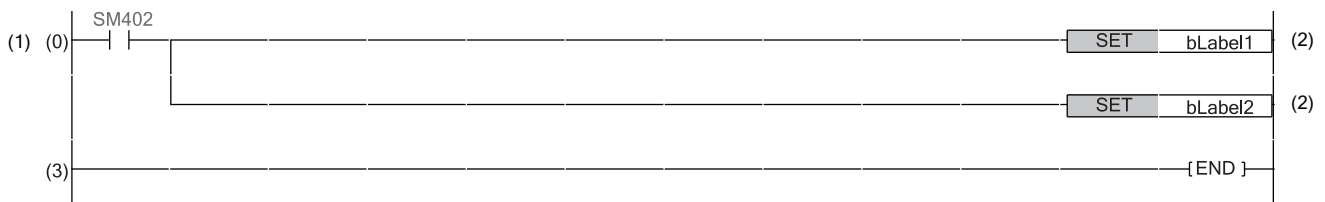
The following shows a program example of matching the operations of function blocks that operate differently from PX Developer with the operations of PX Developer.

## Program example

Operations related to the public variable IR in Bistable Switch (Set Dominant) and Bistable Switch (Reset Dominant) become equal to the corresponding operations of PX Developer by writing the following programs.

The following programs are applicable to cases in which one FB instance for Bistable Switch (Set Dominant) and Bistable Switch (Reset Dominant) each is defined as a local label.

### ■Detection of the start-up of the CPU module (powered off and on, reset)



(1) Special relay to detect the start-up of the CPU module

(2) Global label to hold the detection of the start-up of the CPU module (BOOL)

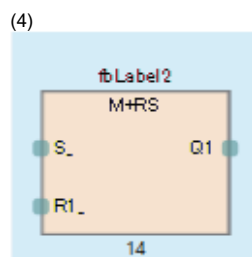
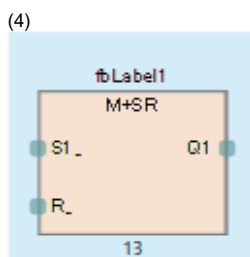
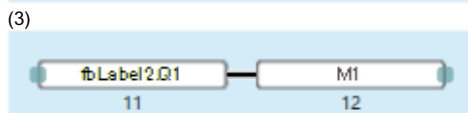
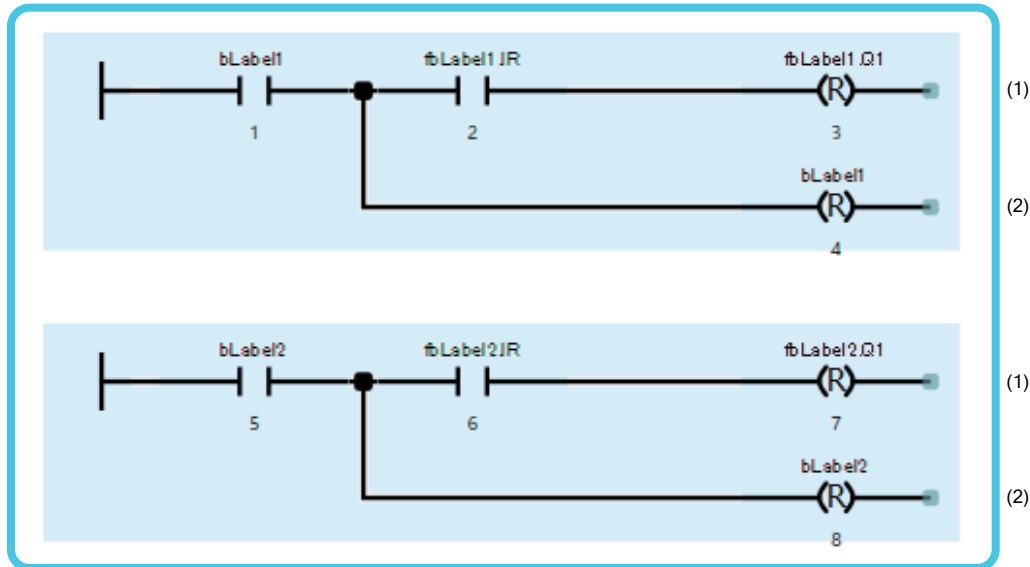
#### Point

- Because the start-up of the CPU module needs to be detected, write the above program in a scan execution type program file and execute them at each scan.
- Write the above program so that it is executed earlier than an output variable Q1 reset processing program that is shown subsequently. (☞ Page 948 Output variable Q1 reset processing)
- When multiple FB instances of Bistable Switch (Set Dominant)/Bistable Switch (Reset Dominant) exist, use a number of global labels (2) equal to the number of FB instances.
- When replacing global labels (2) with global devices in a redundant system, limit them within the tracking transfer setting range.

A

## ■Output variable Q1 reset processing

Add programs (1) and (2) enclosed in a light blue frame to a customer's programs (3) and (4).



- (1) When the start-up of the CPU module is detected (bLabel1 or bLabel2 is TRUE), the output variable Q1 is reset according to the public variable IR.
- (2) The detection of the start-up of the CPU module is reset.
- (3) Output variable Q1 is referred to.
- (4) Bistable Switch (Set Dominant) or Bistable Switch (Reset Dominant) is executed.

### Point

- When multiple FB instances of Bistable Switch (Set Dominant)/Bistable Switch (Reset Dominant) exist, write a number of programs (1) and (2) equal to the number of FB instances.
- Write programs (1) and (2) so that they are executed earlier than the programs (3) reference to the output variable Q1 and (4) execution of Bistable Switch (Set Dominant)/Bistable Switch (Reset Dominant).

# Appendix 7 Replacement of Standard Function Blocks

Some standard function blocks can be replaced with standard process function blocks with the latch attributes. The following shows target function blocks and the replacement procedure.

## Target function blocks

The following table lists source standard function blocks and target standard process function blocks.

Standard function block	Standard process function block
SR	M+SR
RS	M+RS
LATCH_BOOL_PX	M+LATCH_BOOL
LATCH_REAL_PX	M+LATCH_REAL
LATCH_WORD_PX	M+LATCH_WORD
LATCH_DWORD_PX	M+LATCH_DWORD
R_TRIG	M+R_TRIG
F_TRIG	M+F_TRIG
EDGE_CHECK_PX	M+EDGE_CHECK
CTU	M+CTU
CTD	M+CTD
CTUD	M+CTUD
TP_HIGH_PX	M+TP_HIGH
TP_LOW_PX	M+TP_LOW
TON_HIGH_PX	M+TON_HIGH
TON_LOW_PX	M+TON_LOW
TOF_HIGH_PX	M+TOF_HIGH
TOF_LOW_PX	M+TOF_LOW

A

## Replacement procedure

The following shows a replacement procedure.

1. Delete the label of the source standard function block type from the program editor and label editor.
2. Drag and drop the target process standard function block from the Element Selection window to register a label.
3. Connect I/O pins.

# Appendix 8 Version Upgrade

Process control function blocks are upgraded for improving or adding functionality at the time of version upgrade of GX Works3. For this reason, version of a function block used in a project which is created with an earlier GX Works3 version may not be the latest one.

When such a project is opened or read from the programmable controller, users may be prompted to upgrade the version of the process control function block. In this case, check the precautions and perform version upgrade. If a program for process control is changed without upgrading the version of the process control function block, unintended operations may result.

## Point

The version of a process control function block can be updated on the tag FB setting editor. For the operations on the tag FB setting editor, refer to the following.

 GX Works3 Operating Manual

## Precautions

When updating the version of a process control function block, note the following points:

- Converting (reassigning) all programs may be required after an update.
- The initial values of FB properties return to the default after an update. However, the initial values can easily be set again by exporting the values in the "FB Property Management (Offline)" window before updating the version, then importing them after the update.

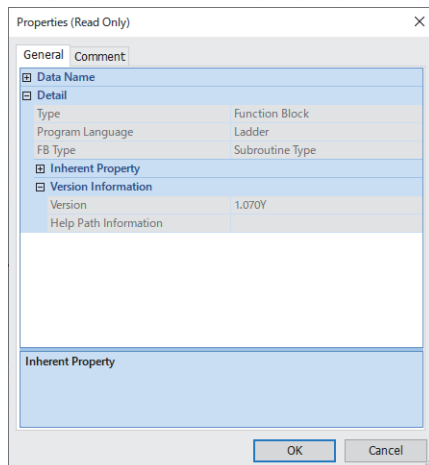
For the operations on the "FB Property Management (Offline)" window, refer to the following.

 GX Works3 Operating Manual

## Checking the version information

The version information can be checked from the properties of the function block.

 [Navigation window] ⇒ [FB/FUN] ⇒ Right-click on the function block's file name ⇒ [Properties]



\*1 A function block whose version is blank is the first version (1.032J).



# Version upgrade history

The functions added/changed with the version upgrade of GX Works3 are listed below.

## Process control function blocks

Name	Version	Addition/change description	Reference
M+P_IN	1.070Y	■Addition of the following public variable: • SEA_OTYPE	Page 184 M+P_IN
	1.085P	■Addition of the following public variable: • LMT_TYPE	
M+P_OUT1	1.085P	■Addition of the following public variable: • SEA_OTYPE	Page 189 M+P_OUT1
M+P_OUT2	1.085P	■Addition of the following public variable: • SEA_OTYPE	Page 194 M+P_OUT2
M+P_DUTY	1.085P	■Addition of the following public variable: • SEA_OTYPE	Page 208 M+P_DUTY
M+P_MCHG	1.070Y	■Addition of the following public variable: • BNAL_MODE	Page 378 M+P_MCHG
M+P_MCHGPRMRY	1.070Y	■Newly added	Page 381 M+P_MCHGPRMRY
M+M_PID_T	1.070Y	■Addition of the following to the constituent function block: • MCHGPRMRY function ■Addition of the following public variables: • IN_SEA_OTYPE • MCHG_BNAL_MODE • MCHGPRMRY_SEA_EN • MCHGPRMRY_OOA_EN • MCHGPRMRY_SPA_EN • MCHGPRMRY_CASCASDR_EN	Page 386 M+M_PID_T
	1.085P	■Addition of the following public variables: • IN_LMT_TYPE • OUT1_SEA_OTYPE	
M+M_PID	1.070Y	■Addition of the following public variables: • IN_SEA_OTYPE • MCHG_BNAL_MODE	Page 390 M+M_PID
	1.085P	■Addition of the following public variables: • IN_LMT_TYPE • OUT1_SEA_OTYPE	
M+M_PID_DUTY_T	1.070Y	■Addition of the following to the constituent function block: • MCHGPRMRY function ■Addition of the following public variables: • IN_SEA_OTYPE • MCHG_BNAL_MODE • MCHGPRMRY_SEA_EN • MCHGPRMRY_OOA_EN • MCHGPRMRY_SPA_EN • MCHGPRMRY_CASCASDR_EN	Page 393 M+M_PID_DUTY_T
	1.085P	■Addition of the following public variables: • IN_LMT_TYPE • DUTY_SEA_OTYPE	
M+M_PID_DUTY	1.070Y	■Addition of the following public variables: • IN_SEA_OTYPE • MCHG_BNAL_MODE	Page 396 M+M_PID_DUTY
	1.085P	■Addition of the following public variables: • IN_LMT_TYPE • DUTY_SEA_OTYPE	

Name	Version	Addition/change description	Reference
M+M_2PID_T	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following to the constituent function block: <ul style="list-style-type: none"> <li>• MCHGPRMRY function</li> </ul> </li> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> <li>• MCHGPRMRY_SEA_EN</li> <li>• MCHGPRMRY_OOA_EN</li> <li>• MCHGPRMRY_SPA_EN</li> <li>• MCHGPRMRY_CASCASDR_EN</li> </ul> </li> </ul>	Page 399 M+M_2PID_T
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> <li>• OUT1_SEA_OTYPE</li> </ul> </li> </ul>	
M+M_2PID	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> </ul> </li> </ul>	Page 402 M+M_2PID
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> <li>• OUT1_SEA_OTYPE</li> </ul> </li> </ul>	
M+M_2PID_DUTY_T	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following to the constituent function block: <ul style="list-style-type: none"> <li>• MCHGPRMRY function</li> </ul> </li> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> <li>• MCHGPRMRY_SEA_EN</li> <li>• MCHGPRMRY_OOA_EN</li> <li>• MCHGPRMRY_SPA_EN</li> <li>• MCHGPRMRY_CASCASDR_EN</li> </ul> </li> </ul>	Page 405 M+M_2PID_DUTY_T
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> <li>• DUTY_SEA_OTYPE</li> </ul> </li> </ul>	
M+M_2PID_DUTY	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> </ul> </li> </ul>	Page 408 M+M_2PID_DUTY
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> <li>• DUTY_SEA_OTYPE</li> </ul> </li> </ul>	
M+M_2PIDH_T_	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following to the constituent function block: <ul style="list-style-type: none"> <li>• MCHGPRMRY function</li> </ul> </li> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> <li>• MCHGPRMRY_SEA_EN</li> <li>• MCHGPRMRY_OOA_EN</li> <li>• MCHGPRMRY_SPA_EN</li> <li>• MCHGPRMRY_TSTP_EN</li> <li>• MCHGPRMRY_CASCASDR_EN</li> </ul> </li> </ul>	Page 411 M+M_2PIDH_T_
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variable: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> </ul> </li> </ul>	
M+M_2PIDH_	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> </ul> </li> </ul>	Page 417 M+M_2PIDH_
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variable: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> </ul> </li> </ul>	

Name	Version	Addition/change description	Reference
M+M_PIDP_T	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following to the constituent function block: <ul style="list-style-type: none"> <li>• MCHGPRMRY function</li> </ul> </li> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> <li>• MCHGPRMRY_SEA_EN</li> <li>• MCHGPRMRY_OOA_EN</li> <li>• MCHGPRMRY_SPA_EN</li> <li>• MCHGPRMRY_CASCASDR_EN</li> </ul> </li> </ul>	Page 424 M+M_PIDP_T
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variable: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> </ul> </li> </ul>	
M+M_PIDP	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> </ul> </li> </ul>	Page 427 M+M_PIDP
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variable: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> </ul> </li> </ul>	
M+M_PIDP_EX_T_	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following to the constituent function block: <ul style="list-style-type: none"> <li>• MCHGPRMRY function</li> </ul> </li> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> <li>• MCHGPRMRY_SEA_EN</li> <li>• MCHGPRMRY_OOA_EN</li> <li>• MCHGPRMRY_SPA_EN</li> <li>• MCHGPRMRY_CASCASDR_EN</li> </ul> </li> </ul>	Page 430 M+M_PIDP_EX_T_
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variable: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> </ul> </li> </ul>	
M+M_PIDP_EX_	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> </ul> </li> </ul>	Page 433 M+M_PIDP_EX_
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variable: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> </ul> </li> </ul>	
M+M_SPI_T	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following to the constituent function block: <ul style="list-style-type: none"> <li>• MCHGPRMRY function</li> </ul> </li> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> <li>• MCHGPRMRY_SEA_EN</li> <li>• MCHGPRMRY_OOA_EN</li> <li>• MCHGPRMRY_SPA_EN</li> <li>• MCHGPRMRY_CASCASDR_EN</li> </ul> </li> </ul>	Page 436 M+M_SPI_T
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> <li>• OUT1_SEA_OTYPE</li> </ul> </li> </ul>	
M+M_SPI	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> </ul> </li> </ul>	Page 439 M+M_SPI
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> <li>• OUT1_SEA_OTYPE</li> </ul> </li> </ul>	
M+M_IPD_T	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following to the constituent function block: <ul style="list-style-type: none"> <li>• MCHGPRMRY function</li> </ul> </li> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> <li>• MCHGPRMRY_SEA_EN</li> <li>• MCHGPRMRY_OOA_EN</li> <li>• MCHGPRMRY_SPA_EN</li> <li>• MCHGPRMRY_CASCASDR_EN</li> </ul> </li> </ul>	Page 442 M+M_IPD_T
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> <li>• OUT1_SEA_OTYPE</li> </ul> </li> </ul>	



Name	Version	Addition/change description	Reference
M+M_IPD	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> </ul> </li> </ul>	Page 445 M+M_IPD
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> <li>• OUT1_SEA_OTYPE</li> </ul> </li> </ul>	
M+M_BPI_T	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following to the constituent function block: <ul style="list-style-type: none"> <li>• MCHGPRMRY function</li> </ul> </li> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> <li>• MCHGPRMRY_SEA_EN</li> <li>• MCHGPRMRY_OOA_EN</li> <li>• MCHGPRMRY_SPA_EN</li> <li>• MCHGPRMRY_CASCASDR_EN</li> </ul> </li> </ul>	Page 448 M+M_BPI_T
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> <li>• OUT1_SEA_OTYPE</li> </ul> </li> </ul>	
M+M_BPI	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> </ul> </li> </ul>	Page 451 M+M_BPI
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> <li>• OUT1_SEA_OTYPE</li> </ul> </li> </ul>	
M+M_R_T	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following to the constituent function block: <ul style="list-style-type: none"> <li>• MCHGPRMRY function</li> </ul> </li> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> <li>• MCHGPRMRY_SEA_EN</li> <li>• MCHGPRMRY_OOA_EN</li> <li>• MCHGPRMRY_SPA_EN</li> <li>• MCHGPRMRY_CASCASDR_EN</li> </ul> </li> </ul>	Page 454 M+M_R_T
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> <li>• OUT2_SEA_OTYPE</li> </ul> </li> </ul>	
M+M_R	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> </ul> </li> </ul>	Page 457 M+M_R
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> <li>• OUT2_SEA_OTYPE</li> </ul> </li> </ul>	
M+M_ONF2_T	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following to the constituent function block: <ul style="list-style-type: none"> <li>• MCHGPRMRY function</li> </ul> </li> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> <li>• MCHGPRMRY_SEA_EN</li> <li>• MCHGPRMRY_OOA_EN</li> <li>• MCHGPRMRY_SPA_EN</li> <li>• MCHGPRMRY_CASCASDR_EN</li> </ul> </li> </ul>	Page 460 M+M_ONF2_T
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variable: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> </ul> </li> </ul>	
M+M_ONF2	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> </ul> </li> </ul>	Page 463 M+M_ONF2
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variable: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> </ul> </li> </ul>	

Name	Version	Addition/change description	Reference
M+M_ONF3_T	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following to the constituent function block: <ul style="list-style-type: none"> <li>• MCHGPRMRY function</li> </ul> </li> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> <li>• MCHGPRMRY_SEA_EN</li> <li>• MCHGPRMRY_OOA_EN</li> <li>• MCHGPRMRY_SPA_EN</li> <li>• MCHGPRMRY_CASCASDR_EN</li> </ul> </li> </ul>	Page 466 M+M_ONF3_T
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variable: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> </ul> </li> </ul>	
M+M_ONF3	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> </ul> </li> </ul>	Page 469 M+M_ONF3
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variable: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> </ul> </li> </ul>	
M+M_MONI	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following public variable: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> </ul> </li> </ul>	Page 472 M+M_MONI
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variable: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> </ul> </li> </ul>	
M+M_MWM	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> </ul> </li> </ul>	Page 474 M+M_MWM
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variable: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> </ul> </li> </ul>	
M+M_SEL	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following public variable: <ul style="list-style-type: none"> <li>• MCHG_BNAL_MODE</li> </ul> </li> </ul>	Page 480 M+M_SEL
M+M_SEL_T1	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following to the constituent function block: <ul style="list-style-type: none"> <li>• MCHGPRMRY function</li> </ul> </li> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• MCHG_BNAL_MODE</li> <li>• MCHGPRMRY_OOA_EN</li> <li>• MCHGPRMRY_SPA_EN</li> <li>• MCHGPRMRY_CASCASDR_EN</li> </ul> </li> </ul>	Page 482 M+M_SEL_T1
M+M_SEL_T2	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following to the constituent function block: <ul style="list-style-type: none"> <li>• MCHGPRMRY function</li> </ul> </li> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• MCHG_BNAL_MODE</li> <li>• MCHGPRMRY_OOA_EN</li> <li>• MCHGPRMRY_SPA_EN</li> <li>• MCHGPRMRY_CASCASDR_EN</li> </ul> </li> </ul>	Page 484 M+M_SEL_T2
M+M_SEL_T3_	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following to the constituent function block: <ul style="list-style-type: none"> <li>• MCHGPRMRY function</li> </ul> </li> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• MCHG_BNAL_MODE</li> <li>• MCHGPRMRY_OOA_EN</li> <li>• MCHGPRMRY_SPA_EN</li> <li>• MCHGPRMRY_CASCASDR_EN</li> </ul> </li> </ul>	Page 487 M+M_SEL_T3_
M+M_MOUT	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following public variable: <ul style="list-style-type: none"> <li>• MCHG_BNAL_MODE</li> </ul> </li> </ul>	Page 490 M+M_MOUT
M+M_PGS	1.070Y	<ul style="list-style-type: none"> <li>■ Version upgrade of the following constituent function block: <ul style="list-style-type: none"> <li>• MCHG function (version 1.070Y)</li> </ul> </li> </ul>	Page 492 M+M_PGS
M+M_PGS2_	1.070Y	<ul style="list-style-type: none"> <li>■ Version upgrade of the following constituent function block: <ul style="list-style-type: none"> <li>• MCHG function (version 1.070Y)</li> </ul> </li> </ul>	Page 494 M+M_PGS2_
M+M_SWM_	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> </ul> </li> </ul>	Page 498 M+M_SWM_
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variable: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> </ul> </li> </ul>	



Name	Version	Addition/change description	Reference
M+M_PVAL_T_	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following to the constituent function block: <ul style="list-style-type: none"> <li>• MCHGPRMRY function</li> </ul> </li> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• IN_SEA_OTYPE</li> <li>• MCHG_BNAL_MODE</li> <li>• MCHGPRMRY_SEA_EN</li> <li>• MCHGPRMRY_SPA_EN</li> <li>• MCHGPRMRY_TSTP_EN</li> <li>• MCHGPRMRY_CASCASDR_EN</li> </ul> </li> </ul>	Page 501 M+M_PVAL_T_
	1.085P	<ul style="list-style-type: none"> <li>■ Addition of the following public variable: <ul style="list-style-type: none"> <li>• IN_LMT_TYPE</li> </ul> </li> </ul>	
M+M_HTCL_T_	1.070Y	<ul style="list-style-type: none"> <li>■ Addition of the following to the constituent function block: <ul style="list-style-type: none"> <li>• MCHGPRMRY function</li> </ul> </li> <li>■ Addition of the following public variables: <ul style="list-style-type: none"> <li>• MCHG_BNAL_MODE</li> <li>• MCHGPRMRY_OOA_EN</li> <li>• MCHGPRMRY_SPA_EN</li> <li>• MCHGPRMRY_TSTP_EN</li> <li>• MCHGPRMRY_CASCASDR_EN</li> </ul> </li> </ul>	Page 510 M+M_HTCL_T_
M+M_NREV	1.070Y	<ul style="list-style-type: none"> <li>■ Version upgrade of the following constituent function block: <ul style="list-style-type: none"> <li>• MCHG function (version 1.070Y)</li> </ul> </li> </ul>	Page 520 M+M_NREV
M+M_REV	1.070Y	<ul style="list-style-type: none"> <li>■ Version upgrade of the following constituent function block: <ul style="list-style-type: none"> <li>• MCHG function (version 1.070Y)</li> </ul> </li> </ul>	Page 524 M+M_REV
M+M_MVAL1	1.070Y	<ul style="list-style-type: none"> <li>■ Version upgrade of the following constituent function block: <ul style="list-style-type: none"> <li>• MCHG function (version 1.070Y)</li> </ul> </li> </ul>	Page 528 M+M_MVAL1
M+M_MVAL2	1.070Y	<ul style="list-style-type: none"> <li>■ Version upgrade of the following constituent function block: <ul style="list-style-type: none"> <li>• MCHG function (version 1.070Y)</li> </ul> </li> </ul>	Page 532 M+M_MVAL2
M+M_PB_	1.070Y	<ul style="list-style-type: none"> <li>■ Version upgrade of the following constituent function block: <ul style="list-style-type: none"> <li>• MCHG function (version 1.070Y)</li> </ul> </li> </ul>	Page 546 M+M_PB_
M+SR	1.080J	■ Newly added	Page 155 M+SR
M+RS	1.080J	■ Newly added	Page 157 M+RS
M+LATCH_BOOL	1.080J	■ Newly added	Page 159 M+LATCH_BOOL
M+LATCH_REAL	1.080J	■ Newly added	Page 160 M+LATCH_REAL
M+LATCH_WORD	1.080J	■ Newly added	Page 161 M+LATCH_WORD
M+LATCH_DWORD	1.080J	■ Newly added	Page 162 M+LATCH_DWORD
M+R_TRIG	1.080J	■ Newly added	Page 163 M+R_TRIG
M+F_TRIG	1.080J	■ Newly added	Page 164 M+F_TRIG
M+EDGE_CHECK	1.080J	■ Newly added	Page 165 M+EDGE_CHECK
M+CTU	1.080J	■ Newly added	Page 166 M+CTU
M+CTD	1.080J	■ Newly added	Page 168 M+CTD
M+CTUD	1.080J	■ Newly added	Page 169 M+CTUD
M+TP_HIGH	1.080J	■ Newly added	Page 171 M+TP_HIGH
M+TP_LOW	1.080J	■ Newly added	Page 173 M+TP_LOW
M+TON_HIGH	1.080J	■ Newly added	Page 175 M+TON_HIGH
M+TON_LOW	1.080J	■ Newly added	Page 177 M+TON_LOW
M+TOF_HIGH	1.080J	■ Newly added	Page 179 M+TOF_HIGH
M+TOF_LOW	1.080J	■ Newly added	Page 181 M+TOF_LOW

# INDEX

---

## C

---

Control cycle . . . . .	36
Control mode . . . . .	27

## E

---

Execution cycle. . . . .	31
--------------------------	----

## F

---

Faceplate . . . . .	48
FB Property . . . . .	47
FBD/LD program for process control . . . . .	24

## I

---

I/O mode . . . . .	28
Instruction processing time. . . . .	801

## P

---

Phase . . . . .	32
Process control instructions . . . . .	560

## S

---

Specifying public variables. . . . .	43
Specifying tag data . . . . .	43
Specifying tag data of a user-defined tag FB . . . . .	46

## T

---

Tag FB Setting . . . . .	41
Tag type . . . . .	30

## U

---

User-defined tag FB . . . . .	29
-------------------------------	----



# INSTRUCTION INDEX

---

## E

---

EDGE\_CHECK\_PX . . . . . 945

## L

---

LATCH\_BOOL\_PX . . . . . 945  
LATCH\_DWORD\_PX . . . . . 945  
LATCH\_REAL\_PX . . . . . 945  
LATCH\_WORD\_PX . . . . . 945

## M

---

M+CTD . . . . . 168  
M+CTU . . . . . 166  
M+CTUD . . . . . 169  
M+EDGE\_CHECK . . . . . 165  
M+F\_TRIG . . . . . 164  
M+LATCH\_BOOL . . . . . 159  
M+LATCH\_DWORD . . . . . 162  
M+LATCH\_REAL . . . . . 160  
M+LATCH\_WORD . . . . . 161  
M+M\_2PID . . . . . 402  
M+M\_2PID\_DUTY . . . . . 408  
M+M\_2PID\_DUTY\_T . . . . . 405  
M+M\_2PIDH . . . . . 417  
M+M\_2PIDH\_T . . . . . 411  
M+M\_2PID\_T . . . . . 399  
M+M\_ALARM . . . . . 550  
M+M\_ALARM\_64PT . . . . . 552  
M+M\_BC . . . . . 476  
M+M\_BPI . . . . . 451  
M+M\_BPI\_T . . . . . 448  
M+M\_COUNTER1 . . . . . 540  
M+M\_COUNTER2 . . . . . 543  
M+M\_HTCL\_T . . . . . 510  
M+M\_IPD . . . . . 445  
M+M\_IPD\_T . . . . . 442  
M+M\_MESSAGE . . . . . 554  
M+M\_MESSAGE\_64PT . . . . . 556  
M+M\_MONI . . . . . 472  
M+M\_MOUT . . . . . 490  
M+M\_MVAL1 . . . . . 528  
M+M\_MVAL2 . . . . . 532  
M+M\_MWM . . . . . 474  
M+M\_NREV . . . . . 520  
M+M\_ONF2 . . . . . 463  
M+M\_ONF2\_T . . . . . 460  
M+M\_ONF3 . . . . . 469  
M+M\_ONF3\_T . . . . . 466  
M+M\_PB . . . . . 546  
M+M\_PGS . . . . . 492  
M+M\_PGS2 . . . . . 494  
M+M\_PID . . . . . 390  
M+M\_PID\_DUTY . . . . . 396  
M+M\_PID\_DUTY\_T . . . . . 393  
M+M\_PIDP . . . . . 427  
M+M\_PIDP\_EX . . . . . 433  
M+M\_PIDP\_EX\_T . . . . . 430  
M+M\_PIDP\_T . . . . . 424  
M+M\_PID\_T . . . . . 386  
M+M\_PSUM . . . . . 478

M+M\_PVAL\_T . . . . . 501  
M+M\_R . . . . . 457  
M+M\_REV . . . . . 524  
M+M\_R\_T . . . . . 454  
M+M\_SEL . . . . . 480  
M+M\_SEL\_T1 . . . . . 482  
M+M\_SEL\_T2 . . . . . 484  
M+M\_SEL\_T3 . . . . . 487  
M+M\_SPI . . . . . 439  
M+M\_SPI\_T . . . . . 436  
M+M\_SWM . . . . . 498  
M+M\_TIMER1 . . . . . 536  
M+M\_TIMER2 . . . . . 538  
M+P\_2PID . . . . . 248  
M+P\_2PIDH . . . . . 262  
M+P\_2PIDH\_T . . . . . 254  
M+P\_2PID\_T . . . . . 242  
M+P\_ABS(E) . . . . . 80  
M+P\_ADD . . . . . 103  
M+P\_AMR . . . . . 147  
M+P\_AVE(E) . . . . . 78  
M+P\_BC . . . . . 216  
M+P\_BPI . . . . . 326  
M+P\_BPI\_T . . . . . 321  
M+P\_BUMP . . . . . 145  
M+P\_D . . . . . 129  
M+P\_DBND . . . . . 143  
M+P\_DED . . . . . 131  
M+P\_DIV . . . . . 109  
M+P\_DUTY . . . . . 208  
M+P\_DUTY\_8PT . . . . . 149  
M+P\_ENG . . . . . 90  
M+P\_EQ . . . . . 117  
M+P\_FG . . . . . 82  
M+P\_FLT . . . . . 88  
M+P\_GE . . . . . 119  
M+P\_GT . . . . . 113  
M+P\_HS(E) . . . . . 72  
M+P\_I . . . . . 126  
M+P\_IENG . . . . . 92  
M+P\_IFG . . . . . 85  
M+P\_IN . . . . . 184  
M+P\_IPD . . . . . 316  
M+P\_IPD\_T . . . . . 310  
M+P\_LE . . . . . 121  
M+P\_LIMT . . . . . 134  
M+P\_LLAG . . . . . 123  
M+P\_LS(E) . . . . . 74  
M+P\_LT . . . . . 115  
M+P\_MCHG . . . . . 378  
M+P\_MCHGPRMRY . . . . . 381  
M+P\_MID(E) . . . . . 76  
M+P\_MOUT . . . . . 206  
M+P\_MSET . . . . . 219  
M+P\_MUL . . . . . 107  
M+P\_ONF2 . . . . . 338  
M+P\_ONF2\_T . . . . . 335  
M+P\_ONF3 . . . . . 345  
M+P\_ONF3\_T . . . . . 341  
M+P\_OUT1 . . . . . 189  
M+P\_OUT2 . . . . . 194  
M+P\_OUT3 . . . . . 198



M+P_PGS	348
M+P_PGS2	352
M+P_PHPL	331
M+P_PID	236
M+P_PIDP	276
M+P_PIDP_EX	291
M+P_PIDP_EX_T	283
M+P_PIDP_T	269
M+P_PID_T	230
M+P_PSUM	213
M+P_R	227
M+P_RANGE	101
M+P_R_T	223
M+P_SEL	363
M+P_SEL_T1	366
M+P_SEL_T2	370
M+P_SEL_T3	374
M+P_SPI	305
M+P_SPI_T	299
M+P_SQR	111
M+P_SUB	105
M+P_SUM	96
M+P_SUM2	98
M+P_TPC	94
M+P_VLMT1	137
M+P_VLMT2	140
M+RS	157
M+R_TRIG	163
M+SR	155
M+TOF_HIGH	179
M+TOF_LOW	181
M+TON_HIGH	175
M+TON_LOW	177
M+TP_HIGH	171
M+TP_LOW	173

S.LIMIT	701
S.LLAG	681
S.LS	694
S.MID	696
S.MOUT	600
S.MUL	763
S.ONF2	710
S.ONF3	716
S.OUT1	589
S.OUT2	595
S.PGS	724
S.PHPL	667
S.PHPL2	673
S.PID	618
S.PIDP	632
S.PSUM	613
S.R	662
S.SEL	729
S.SPI	640
S.SQR	767
S.SUB	761
S.SUM	750
S.TPC	752
S.VLMT1	704
S.VLMT2	707
S_2PID	625
S_ABS	769
S_ADD	759
S_AMR	738
S_AT1	784
S_AVE	699
S_BC	609
S_BPI	655
S_BUMP	735
S_D	686
S_DBND	722
S_DED	689
S_DIV	765
S_DUTY	603
S_ENG	755
S_EQ	775
S_FG	741
S_FLT	747
S_GE	777
S_GT	771
S_HS	692
S_I	684
S_IENG	757
S_IFG	744
S_IN	583
S_IPD	648
S_LE	779
S_LIMIT	701
S_LLAG	681
S_LS	694
S_LT	773
S_MID	696
S_MOUT	600
S_MUL	763
S_ONF2	710
S_ONF3	716
S_OUT1	589
S_OUT2	595
S_PGS	724
S_PHPL	667
S_PHPL2	673

## S

S.<	773
S.<=	779
S.=	775
S.>	771
S.>=	777
S.2PID	625
S.ABS	769
S.ADD	759
S.AMR	738
S.AT1	784
S.AVE	699
S.BC	609
S.BPI	655
S.BUMP	735
S.D	686
S.DBND	722
S.DED	689
S.DIV	765
S.DUTY	603
S.ENG	755
S.FG	741
S.FLT	747
S.HS	692
S.I	684
S.IENG	757
S.IFG	744
S.IN	583
S.IPD	648

S_PID	618
S_PIDP	632
S_PSUM	613
S_R	662
S_SEL	729
S_SPI	640
S_SQR	767
S_SUB	761
S_SUM	750
S_TPC	752
S_VLMT1	704
S_VLMT2	707

## T

---

TOF_HIGH_PX	945
TOF_LOW_PX	945
TON_HIGH_PX	945
TON_LOW_PX	945
TP_HIGH_PX	945
TP_LOW_PX	945

# MEMO

---

# REVISIONS

\*The manual number is given on the bottom left of the back cover.

Revision date	*Manual number	Description
January 2017	SH(NA)-081749ENG-A	First edition
March 2017	SH(NA)-081749ENG-B	Partial correction
October 2017	SH(NA)-081749ENG-C	■Added or modified parts Chapter 1, 2, APPENDIX 3, 4
April 2018	SH(NA)-081749ENG-D	■Added or modified parts Chapter 1, 17
April 2019	SH(NA)-081749ENG-E	■Added or modified parts Section 2.5, Chapter 17, Appendix 5
October 2019	SH(NA)-081749ENG-F	Partial correction
October 2020	SH(NA)-081749ENG-G	■Added or modified parts Chapter 1, 4, 10, 12, 13, Appendix 2, 3, 6
December 2020	SH(NA)-081749ENG-H	■Added or modified parts Chapter 1, Appendix 6
October 2021	SH(NA)-081749ENG-I	■Added or modified parts Section 4.1, Chapter 5, 6, 7, 8, 9, 10, 11, 12, 13, Appendix 2, 3, 5, 6, 7
April 2022	SH(NA)-081749ENG-J	■Added or modified parts Section 14.1, 14.2, 14.3, 14.6, 17.1, 17.2, 17.3, 17.4, 17.5, 17.6, 17.7, 17.8, 17.9, 17.10, 17.11, 17.12, 17.13, 17.14, 17.15, 17.16, 17.17, 17.18, 17.19, 17.20, 17.21, 17.22, 17.23, 17.24, 17.25, 17.26, 17.27, 17.28, 17.38, 17.39, Appendix 2, 8

Japanese manual number: SH-081748-L

This manual confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 2017 MITSUBISHI ELECTRIC CORPORATION

# WARRANTY

---

Please confirm the following product warranty details before using this product.

## **1. Gratis Warranty Term and Gratis Warranty Range**

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company.

However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place. Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

- (1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.
- (2) Even within the gratis warranty term, repairs shall be charged for in the following cases.
  1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
  2. Failure caused by unapproved modifications, etc., to the product by the user.
  3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
  4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
  5. Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
  6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
  7. Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

## **2. Onerous repair term after discontinuation of production**

- (1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued. Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.
- (2) Product supply (including repair parts) is not available after production is discontinued.

## **3. Overseas service**

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

## **4. Exclusion of loss in opportunity and secondary loss from warranty liability**

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation to:

- (1) Damages caused by any cause found not to be the responsibility of Mitsubishi.
- (2) Loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products.
- (3) Special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products.
- (4) Replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

## **5. Changes in product specifications**

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

# TRADEMARKS

---

The company names, system names and product names mentioned in this manual are either registered trademarks or trademarks of their respective companies.

In some cases, trademark symbols such as <sup>™</sup> or <sup>®</sup> are not specified in this manual.



SH(NA)-081749ENG-J(2204)

MODEL: R-P-PSFB-E

## **mitsubishi electric corporation**

HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN  
NAGOYA WORKS : 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA, JAPAN

When exported from Japan, this manual does not require application to the  
Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.