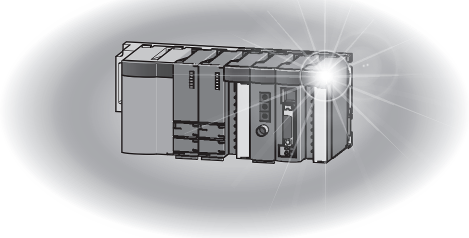
MITSUBISHI ELECTRIC

Programmable Controller

MELSEC Q series

MELSEC-Q
Programming/Structured Programming Manual
(Process Control Instructions)



● SAFETY PRECAUTIONS ●

(Always read these cautions before using the product)

Before using MELSEC-Q series programmable controllers, please read the manuals included with each 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 the manuals included with each product, and keep the manuals 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.

INTRODUCTION

Thank you for purchasing the Mitsubishi MELSEC-Q series programmable controllers.

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

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

Remark

In this manual, instructions are written in three programming languages:
ladder diagram for Simple projects, structured ladder/FBD and structured text language for Structured projects.
Please use GX Works2 with the version 1.98C or later for Structured projects.

CONTENTS

RELEVANT MANUALS

Manual name	Manual number (model code)
QCPU User's Manual (Hardware Design, Maintenance and Inspection) Specifications of the CPU modules, power supply modules, base units, extension cables, memory cards, SD memory cards, extended SRAM cassettes, and batteries, information on how to establish a system, maintenance and inspection, and troubleshooting (Sold separately)	SH-080483ENG (13JR73)
Qn(H)/QnPH/QnPRHCPU User's Manual (Function Explanation, Program Fundamentals) Functions, methods, and devices for programming (Sold separately)	SH-080808ENG (13JZ28)
MELSEC-Q/L Programming Manual (Common Instruction) How to use sequence instructions, basic instructions, and application instructions (Sold separately)	SH-080809ENG (13JW10)
MELSEC-Q/L/QnA Programming Manual (SFC) System configuration, performance specifications, functions, programming, debugging, and error codes for SFC (MELSAP3) programs (Sold separately)	SH-080041 (13JF60)
MELSEC-Q/L Programming Manual (MELSAP-L) Programming methods, specifications, and functions for SFC (MELSAP-L) programs (Sold separately)	SH-080076 (13JF61)
MELSEC-Q/L/F Structured Programming Manual (Fundamentals) Methods and languages for structured programming (Sold separately)	SH-080782ENG (13JW06)
MELSEC-Q/L Structured Programming Manual (Common Instructions) Specifications and functions of common instructions, such as sequence instructions, basic instructions, and application instructions, that can be used in structured programs (Sold separately)	SH-080783ENG (13JW07)

TERMS

Unless otherwise specified, this manual uses the following terms.

Term	Description
QnPHCPU	A generic term for the Q02PHCPU, Q06PHCPU, Q12PHCPU, and Q25PHCPU
QnPRHCPU	A generic term for the Q12PRHCPU and Q25PRHCPU
QnUDPVCPU	A generic term for the Q04UDPVCPU, Q06UDPVCPU, Q13UDPVCPU, and Q26UDPVCPU

Instructions are written in three programming languages:

ladder diagram for Simple projects, structured ladder/FBD and structured text language for Structured projects. To write instructions other than comparison operation instructions in the structured ladder/FBD or structured text language, use "_" instead of ".".

For the comparison operation instructions, use the following instruction symbols.

Instruction	Ladder diagram	Structured ladder/FBD and structured text language
Comparison operation instruction	S.>	S_GT
	S.<	S_LT
	S.=	S_EQ
	S.>=	S_GE
	S.<=	S_LE

CHAPTER 1 OVERVIEW

This manual describes the process control instructions equipped for the CPU module.

1.1 Features

The process control instructions have the following features.

(1) Use of floating-point data

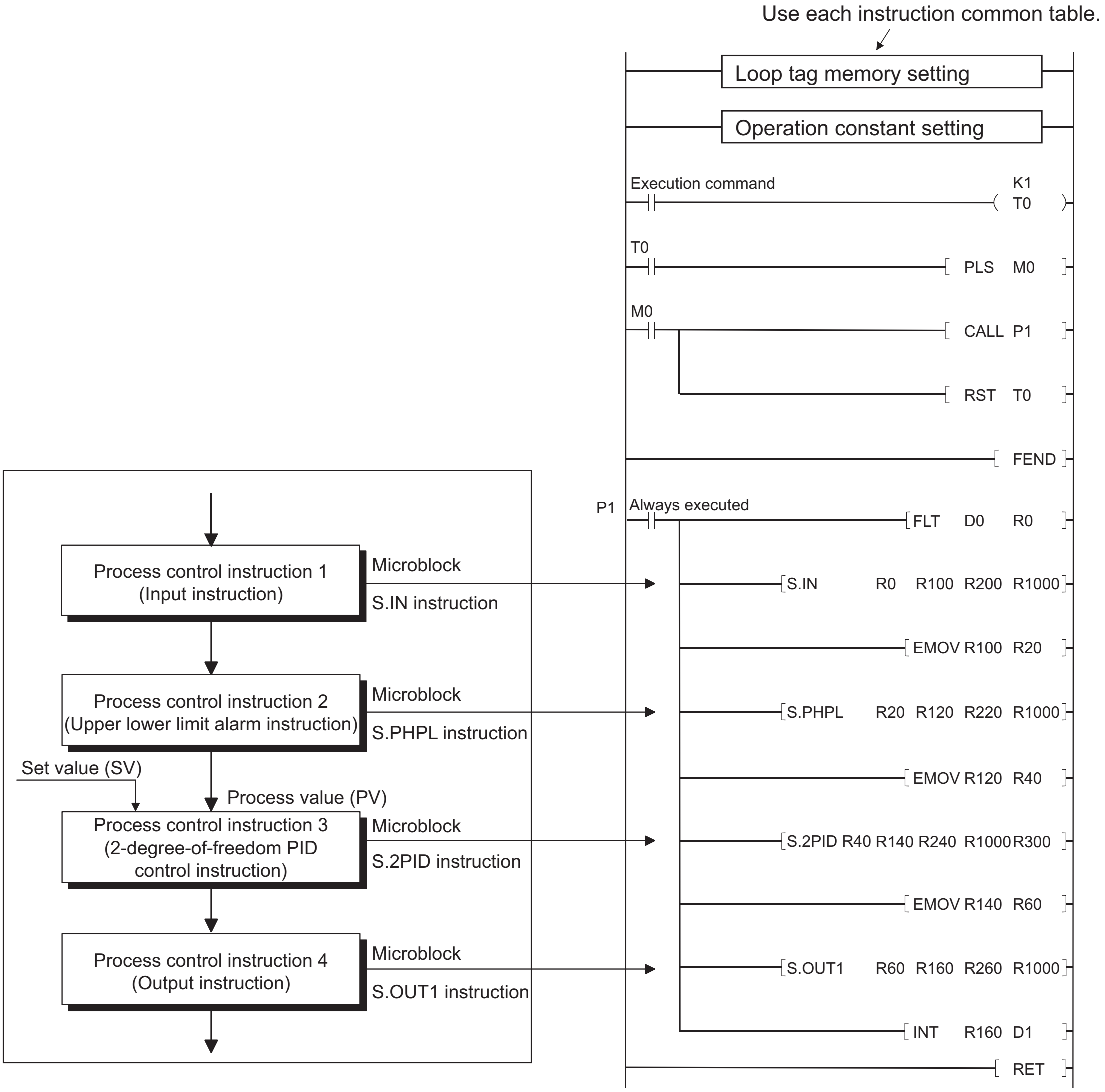
Capable of handling floating-point type real number data, the instructions can perform wide-range and accurate operations.

(2) Increased efficiency of system adjustment

Micro-blocked process control instructions are combined to perform PID control.

This enables actions to be confirmed on a process control instruction basis, ensuring efficient system adjustment.

Example) Process control instructions used to carry out 2-degree-of-freedom PID control

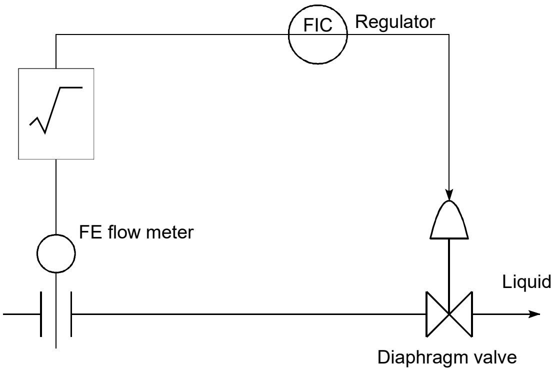


Use each instruction common table.
Loop tag memory setting
Operation constant setting
Execution command
K1
T0
T0
PLS M0
M0
CALL P1
RST T0
FEND
P1
Always executed
FLT D0 R0
S.IN R0 R100 R200 R1000
EMOV R100 R20
S.PHPL R20 R120 R220 R1000
EMOV R120 R40
S.2PID R40 R140 R240 R1000 R300
EMOV R140 R60
S.OUT1 R60 R160 R260 R1000
INT R160 D1
RET
Process control instruction 1 (Input instruction)
Microblock
S.IN instruction
Process control instruction 2 (Upper lower limit alarm instruction)
Microblock
S.PHPL instruction
Set value (SV)
Process value (PV)
Process control instruction 3 (2-degree-of-freedom PID control instruction)
Microblock
S.2PID instruction
Process control instruction 4 (Output instruction)
Microblock
S.OUT1 instruction

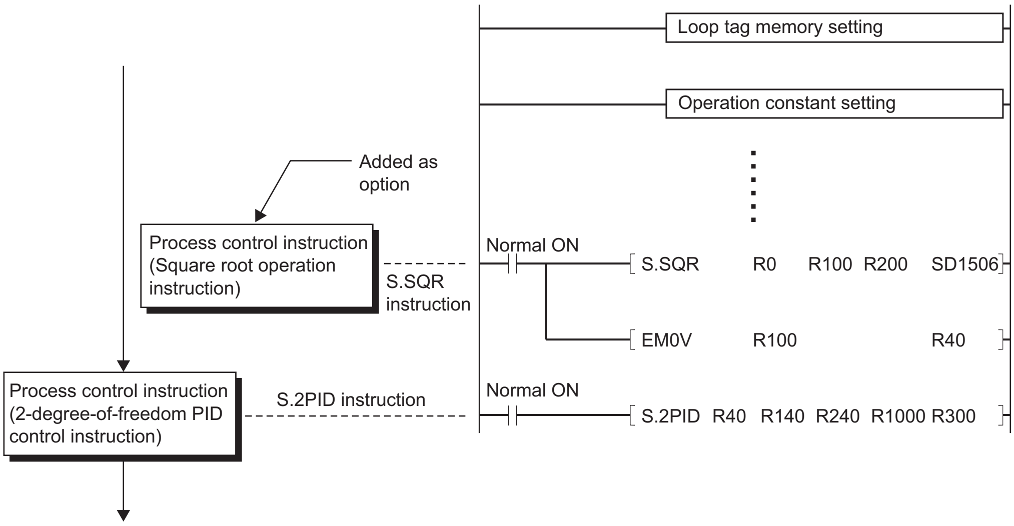
(3) Free combination of process control instructions for application to a wide range of control

As an option, a process control instruction can be inserted in a loop that links process control instructions.

Add the square root operation instruction (S.SQR) to perform the square root operation of an input signal to provide an output signal as shown below.



[Example of adding square root operation instruction (S.SQR) to process control instructions]


Loop tag memory setting
Operation constant setting
Added as option
Process control instruction (Square root operation instruction)
S.SQR instruction
Normal ON
S.SQR R0 R100 R200 SD1506
EM0V R100 R40
Process control instruction (2-degree-of-freedom PID control instruction)
S.2PID instruction
Normal ON
S.2PID R40 R140 R240 R1000 R300

(4) Automatic detection of various alarms

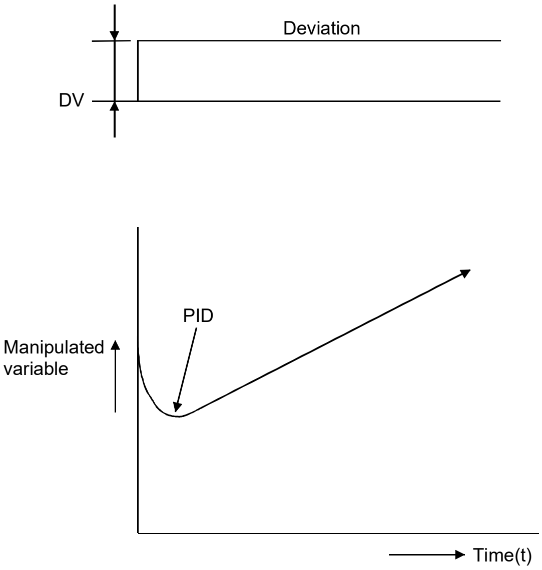
A system can be configured safely since various alarms are detected automatically in the system.

(5) PID algorithm using a velocity type incomplete differential format

Partial differential has the following advantages over the complete differential format.

(a) The differential gain is 1/η and the limit value can be set.

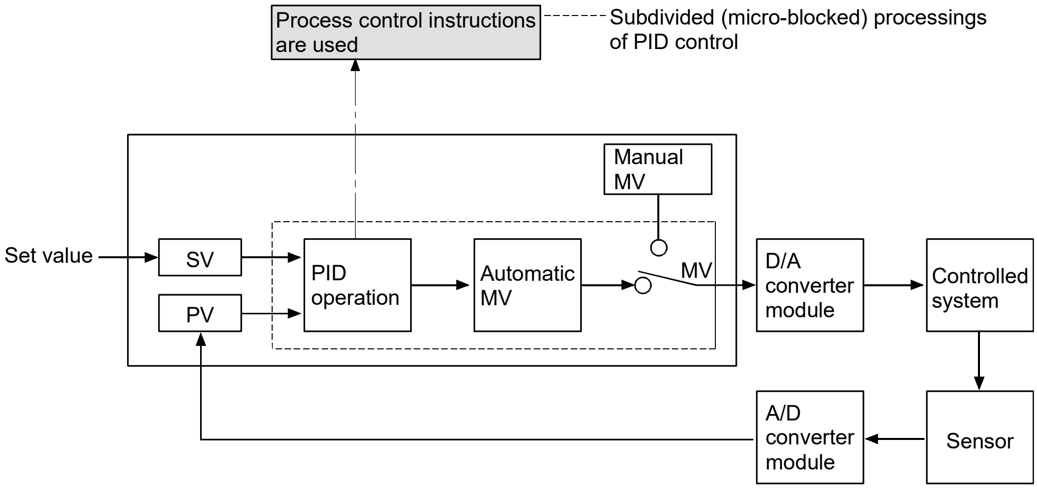
(b) The output contains time amplitude, so the system actually responds to the operation edge so the derivative operation makes the movement valid.



1.2 PID Control Overview

PID control is applied to the process control of flow rate, speed, air volume, temperature, tension, compounding or like.

In the following application, a value of a control target system can be kept at a set value with PID control.



PID control compares the value measured in the detection section (process value: PV) with the preset value (set value: SV) and adjust the output value (manipulated value: MV) to eliminate the difference between the process value and set value.

In PID control, proportional operation (P), integral operation (I) and derivative operation (D) are combined to calculate the manipulated value that will make the process value equal to the set value fast and precisely.

- If the difference between the process value and set value is large, the manipulated value is increased to make it close to the set value fast.
- When the difference between the process value and set value has reduced, the manipulated value is decreased to make it equal to the set value slowly and precisely.

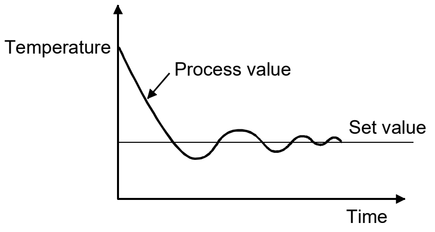
1.3 Forward Operation and Reverse Operation

(1) Forward operation is the action that increases the manipulated value when the process value increases more than the set value.

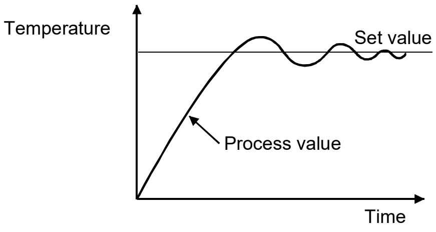
(2) Reverse operation is the action that increases the manipulated value when the process value is decreasing more than the set value.

(3) Forward operation and reverse operation make the manipulated value larger as the difference between the set value and the process value becomes larger.

(4) The following figure shows an example of process control in forward operation and reverse operation.



Forward operation (for air conditioning)



Reverse operation (for heating)

1.4 PID Control Details

This section explains "proportional operation", "integral operation" and "derivative operation" performed for PID control using the process control instructions.

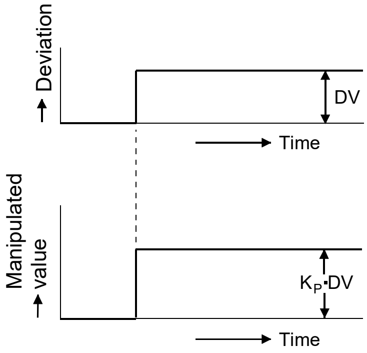
1.4.1 Proportional operation (P operation)

This section explains the control method using proportional operation.

(1) Proportional operation is the action that compares the deviation (DV, difference between the set value and the process value) to find the manipulated value (MV).

(2) The proportional term is given by:
$MV = K_P \cdot DV$
K_P is a proportional gain constant.

(3) The proportional operation in the case of a step response with a constant deviation will be as follows.



(4) The manipulated value fluctuates between -10 and 110%.
As K_P increases, the manipulated value for the constant deviation also increases.

(5) Offset occurs in proportional operation.

1.4.2 Integral operation (I operation)

This section explains the control method using integral operation.

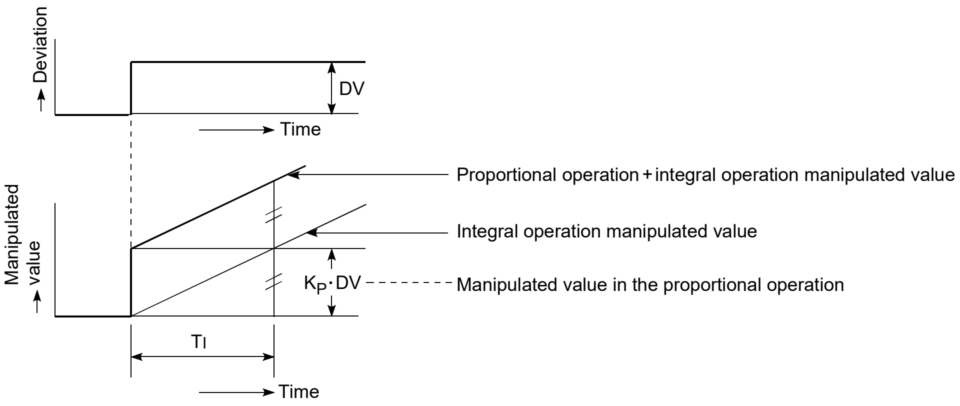
(1) Integral operation is the operation that continuously changes the manipulated value to eliminate deviation when there is deviation.
This operation can eliminate the offset that occurs during control performed by a proportional operation.

(2) The time required for adjusting the manipulated value of the integral operation to the manipulated value of the proportional operation after the deviation is detected is called integral time (T_I).

(a) Increasing the integral time decreases the effect of integration.
(It will take time to stabilize.)

(b) Decreasing the integral time increases the effect of integration.
However, since the integral operation will be stronger, hunting may become greater.

(3) The integral operation in the case of a step response with a constant deviation will be as follows.


Deviation
DV
Time
Manipulated value
Proportional operation + integral operation manipulated value
Integral operation manipulated value
KP·DV
Manipulated value in the proportional operation
TI
Time

(4) The integral operation is used as the PI operation that is combined with the proportional operation or as the PID operation that is combined with the proportional operation and the derivative operation.
Control cannot be carried out by merely performing the integral operation.

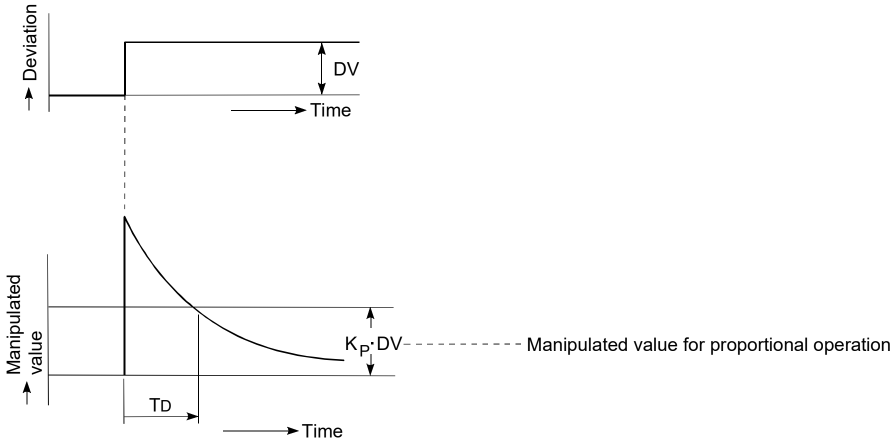
1.4.3 Derivative operation (D operation)

This section explains the control method using the derivative operation.

(1) The derivative operation is an operation that adds the proportional manipulated value to the change speed to eliminate deviation when a deviation has occurred.
The derivative operation can prevent large changes in the object control from disturbances.

(2) The time required for adjusting the manipulated value of the derivative operation to the manipulated value of the proportional operation after the deviation is detected is called derivative time (T_D).
Increasing the derivative time makes the derivative operation stronger.

(3) The derivative operation in the case of a step response with a constant deviation will be as follows.



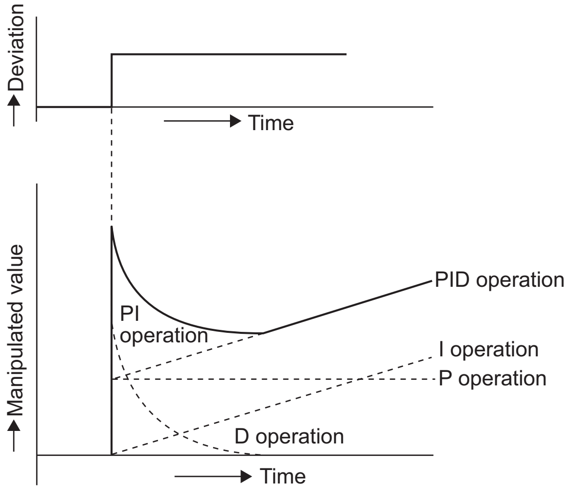
(4) The derivative operation can be used as PD operation in combination with a proportional operation or as a PID operation in combination with the proportional operation and integral operation.
Control cannot be carried out by merely performing the derivative operation.

1.4.4 PID operation

This section explains the control operation using combinations of proportional operation (P operation), integral operation (I operation), and derivative operation (D operation).

(1) The PID operation controls the calculated manipulated value using (P + I + D) operation.

(2) The PID operation in the case of a step response with a constant deviation will be as follows.



CHAPTER 2 STRUCTURE AND COMBINATIONS OF PROCESS CONTROL INSTRUCTIONS

2.1 Structure of Instructions

The instructions that can be used by the process control instructions can be divided into the "instruction part" and "device part".

The instruction part and device part are as follows.

- Instruction part.......This shows the functions for these instructions.
- Device part............This shows the data required for operations and the storage destination of the stored operation results.

The device part is classified as the source device and destination device.

(1) Source (S)

The source stores the data used for operation.

(a) In the process control instruction, specify the start device that stores the source data.

(b) Data must have been stored in the specified device until the process control instruction is executed.

(c) Changing the source data allows you to change the data used in that instruction.

(2) Destination (D)

Destination is where the data is stored after operation.

(a) Sets the device for which the data will be stored in the destination.

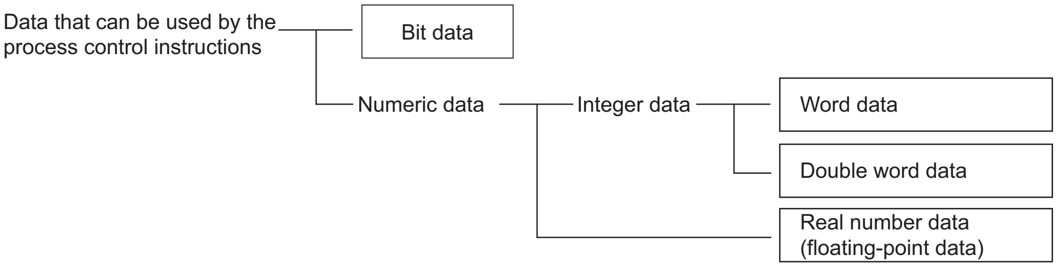
(b) Depending on the instruction used, data used for operation must also be stored in the destination before start of the operation.

Point

For the structure of instructions used in structured programs, refer to the MELSEC-Q/L Structured Programming Manual (Common Instructions).

2.2 How to Specify Data in Devices

The following 4 types of data can be used by the process control instructions.

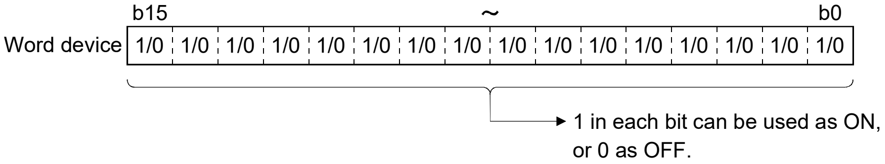

Data that can be used by the process control instructions
Bit data
Numeric data
Integer data
Word data
Double word data
Real number data (floating-point data)

2.2.1 In the case of bit data

Bit data is handled on a single bit basis.

The CPU module uses a word device for alarm condition or selection on a single bit basis.

By specifying the bit number of the word device, you can use the 1/0 of the specified bit number as bit data.



Specify the bit of the word device in the form of "Word device . Bit No. ".

(Specify the bit number in hexadecimal.)

For example, specify the bit 5 (b5) of D0 as D0.5, and the bit 10 (b10) of D0 as D0.A.

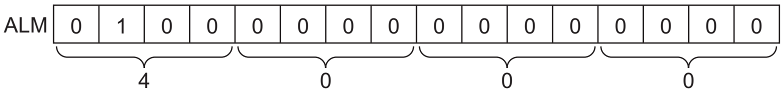
However, you cannot specify the bits of the timer (T), retentive timer (ST), counter (C) and index register (Z). (Example: You cannot specify Z0.0.)

2.2.2 In the case of word (16-bit) data

Word data is the 16-bit numeric data that is used for the loop tag memory bit pack contents and operation constants, etc.

- Decimal constant........................K-32768 to K32767
- Hexadecimal constant................H0000 to HFFFF

Example) For the loop tag memory ALM (standard value setting 4000_H)



2.2.3 In the case of double word (32-bit) data

Double word data is 32-bit numeric data.

- Decimal constant......................K-2147483648 to K2147483647
- Hexadecimal constant..............H00000000 to HFFFFFFFF

When using double word data, specify the word device to be used in the lower-order 16 bits.

The 32-bit data is stored into the (specified word device number) and ((specified word device number) + 1).

Example) When D10 is specified for double word data, D10 and D11 are used.

D11	D10
(BW1)H	(BW1)L

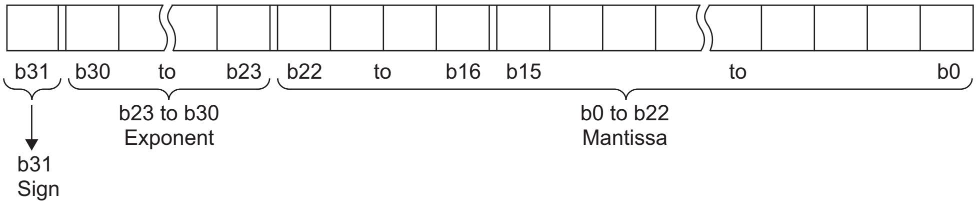
2.2.4 In the case of real number data (floating-point) data

The data required for operations and the operation results are 32-bit floating-point data.

Floating-point data is displayed as follows using 2 word devices.

1. [Fixed-point part] × 2 [Exponent part]

The bit configuration when the floating-point data is expressed internally and its meaning are as follows.



- Fixed-point part sign This shows the fixed-point part sign in b31.
  0: Positive
  1: Negative
- Exponent part This shows the 2^{n}'s n and b23 to b30.
  The n from b23 to b30's BIN value is as follows.

b23 to b30	FF$_{H}$	FE$_{H}$	FD$_{H}$		81$_{H}$	80$_{H}$	7F$_{H}$	7E$_{H}$		02$_{H}$	01$_{H}$	00$_{H}$
n	Non-numeric data	127	126		2	1	0	-1		-125	-126	Non-numeric data

- Fixed-point part This shows the value of XXXXXX... in the 23 bits, b0 to b22, when 1.XXX XXX... is represented in binary.

Point

- The real number setting range is 0, $\pm 2^{-126} \le |\text{value}| < \pm 2^{128}$.
- To represent 0, set 0 in all of b0 to b31.

2.2.5 Operation errors

Operation errors caused by process control instructions are stored in the following remote register.

For errors other than operation errors, refer to the error codes listed in the QCPU User's Manual (Hardware Design, Maintenance and Inspection). (The error code is stored in SD0.)

Remark

(1) The following errors (other than operation errors) are also stored in the special register.
- Error code 4002..........The name of the specified instruction is incorrect.
- Error code 4003..........The number of devices used in the process control instruction is incorrect.
- Error code 4004..........A device that cannot be used in the instruction is specified.
- Error code 4100..........The instruction cannot process the data.

(2) For the error code 4100, the detailed information is stored in special register (SD1502 and SD1503). Values in SD1502 and SD1503 are set to 0 when other than the process control instruction operation error.

For details, refer to Page 254, CHAPTER 14.

2.2.6 Execution conditions

The process control instructions are instructions that are executed while the input condition is ON.

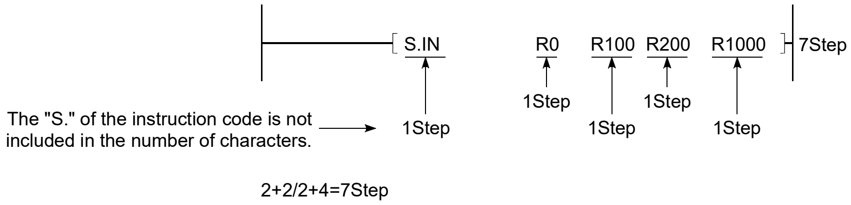
2.2.7 Number of steps

The number of process control instruction steps differs depending upon the number of instruction characters, the device used, and whether or not an indirect setting is valid.

The basic number of steps for the extension instruction are as follows.

$$\text{Number of steps in process control instruction} = 2 + \frac{\text{number of instruction characters}^{*1}}{2} + \text{number of devices}$$

* 1 The number of characters is calculated by adding 1 when the number is odd. (For example when rounding up the results of a division.)

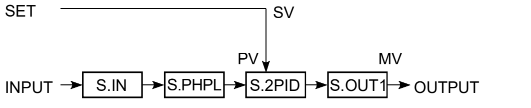
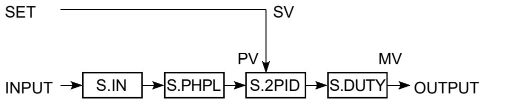
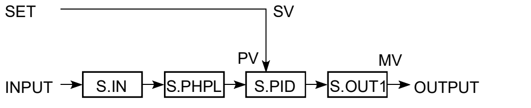
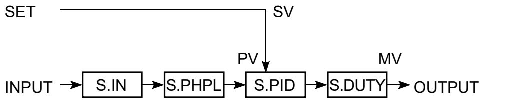
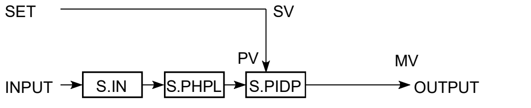
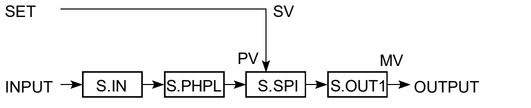
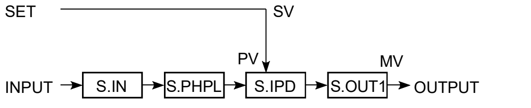
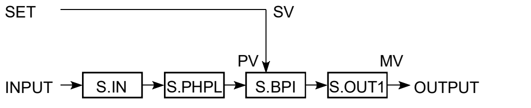
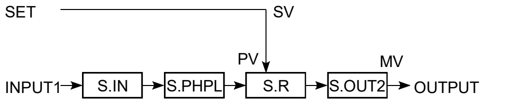
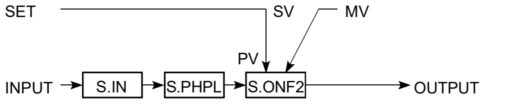
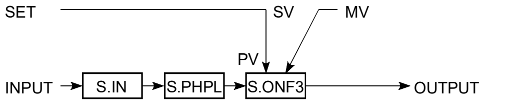
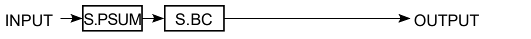

S.IN
R0
R100
R200
R1000
7Step
1Step
1Step
1Step
1Step
1Step
The "S." of the instruction code is not included in the number of characters.
2+2/2+4=7Step

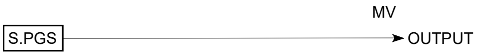
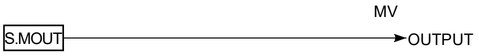
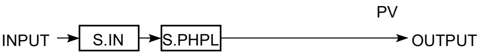
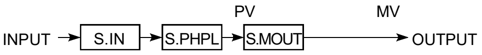
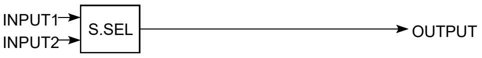
For details, refer to the MELSEC-Q/L Structured Programming Manual (Common Instructions).

2.2.8 Index modification

Index modification available for the process control instructions is the same as the one available for the basic instructions of the CPU module.

2.3 Basic Loop Types Available by Combinations of Process Control Instructions

Loop type	Structure	Application
2-degree-of-freedom PID control (S2PID)	 SET SV PV MV INPUT S.IN S.PHPL S.2PID S.OUT1 OUTPUT  SET SV PV MV INPUT S.IN S.PHPL S.2PID S.DUTY OUTPUT	Used for general PID control (2-degree-of-freedom). (velocity type) Conducts PID operations for each control cycle.
PID control (SPID)	 SET SV PV MV INPUT S.IN S.PHPL S.PID S.OUT1 OUTPUT  SET SV PV MV INPUT S.IN S.PHPL S.PID S.DUTY OUTPUT	Used for general PID control. (velocity type) Conducts PID operations for each control cycle.
PIDP control (SPIDP)	 SET SV PV MV INPUT S.IN S.PHPL S.PIDP OUTPUT	Used for general PID control. (Position type) Conducts PID operation for each control cycle.
Sample PI control (SSPI)	 SET SV PV MV INPUT S.IN S.PHPL S.SPI S.OUT1 OUTPUT	Used for a process that has long dead time. The operation to perform PI control for a set operating time period and then stabilize the output is repeated every sample cycle.
I-PD control (SIPD)	 SET SV PV MV INPUT S.IN S.PHPL S.IPD S.OUT1 OUTPUT	Used to make slow response so that the operation end and process are not given impact when the set value is varied.
Blend PI control (SBPI)	 SET SV PV MV INPUT S.IN S.PHPL S.BPI S.OUT1 OUTPUT	Used for a process where the manipulated value may vary in a short period of time and may be constant in a long period of time.
Ratio control (SR)	 SET SV PV MV INPUT1 S.IN S.PHPL S.R S.OUT2 OUTPUT	Control is performed to keep constant the ratio of the given manipulated value to the other varying value.
2-position ON/OFF control (SONF2)	 SET SV MV PV INPUT S.IN S.PHPL S.ONF2 OUTPUT	Depending on the sign (positive/negative) of a deviation, operation to turn the manipulated value ON or OFF is performed.
3-position ON/OFF control (SONF3)	 SET SV MV PV INPUT S.IN S.PHPL S.ONF3 OUTPUT	3-position ON/OFF control outputs signals of three areas in response to the process value to carry out control. This control can suppress the sudden variation of the manipulated value.
Batch counter (SBC)		A valve or like is controlled ON/OFF in a process of batch preparation for a tank or like.

Loop type	Structure	Application
Program setting device (SPGS)		This is output in accordance with the previously set value time change.
Manual output (SMOUT)		This manually operates the operation terminal end.
Monitor (SMON)		This inputs the process value and detects process errors such as upper/ lower limit alarms.
Manual output with monitor (SMWM)		This inputs the process value and conducts manual operation while checking that no errors occur.
Selector (SSEL)		This is used to select signals.

CHAPTER 3 DATA USED FOR PROCESS CONTROL INSTRUCTIONS AND HOW TO SPECIFY DATA

3.1 Process Control Instructions and Data Structure

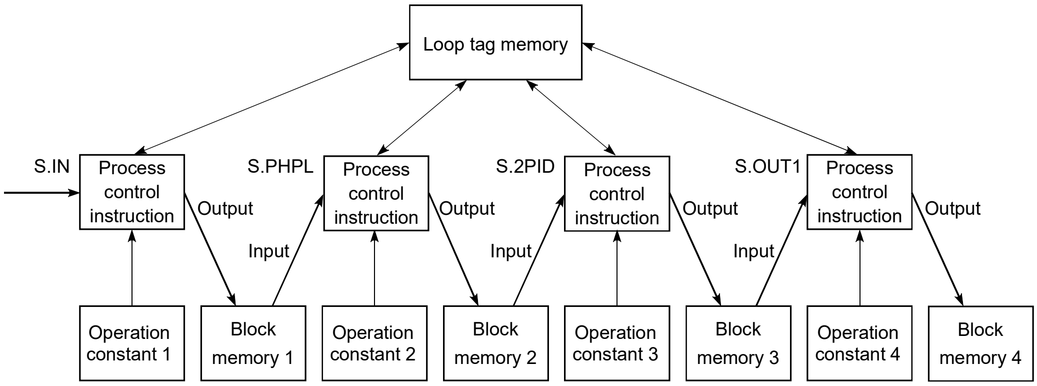
This section explains the data structure (data flow) used for process control instructions.

(a) Configuration when using loop tag

1) The loop units have common storage areas that show the control information. This collection of common information is called a loop tag and the storage memory is called the loop tag memory.

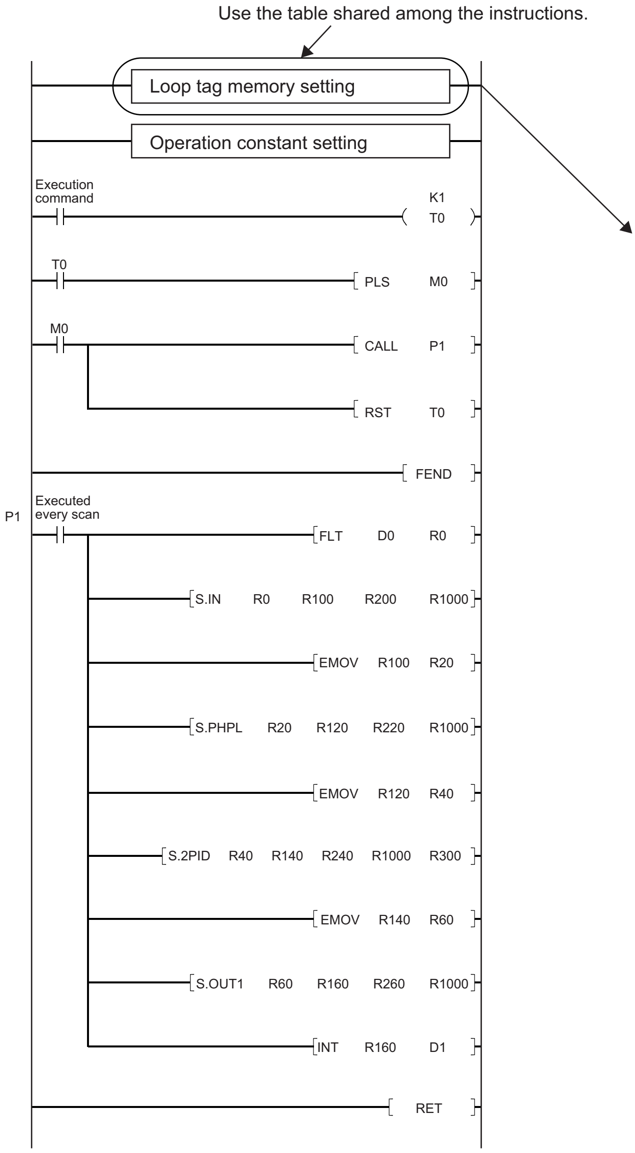
2) By monitoring the loop tag, you can monitor and tune the loop (control unit).

[Block diagram]



(b) Loop tag memory and operation constant locations in ladder diagram

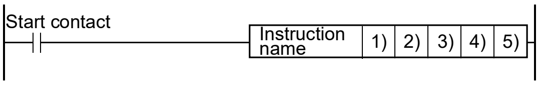
[Ladder diagram]



Loop tag memory (96 words)

	Instruction used	Item	Standard value setting	Data type
+0				BIN16bit
+1		MODE	8_H	BIN16bit
+3		ALM	4000_H	BIN16bit
+4		INH	0_H	BIN16bit
+10	S.PHPL	PV	0.0	Real number
+12	S.OUT1	MV	0.0	Real number
+14	S.2PID	SV	0.0	Real number
+16	S.2PID	DV	0.0	Real number
+18	S.OUT1	MH	100.0	Real number
+20	S.OUT1	ML	0.0	Real number
+22	S.PHPL	RH	100.0	
+46	S.2PID	CT	1.0	Real number
+48	S.OUT1	DML	100.0	Real number
+50	S.2PID	DVL	100.0	Real number
+52	S.2PID	P	1.0	Real number
+54	S.2PID	I	10.0	Real number
+56	S.2PID	D	0.0	Real number
+58	S.2PID	GW	0.0	Real number
+60	S.2PID	GG	1.0	Real number
+62	S.OUT1	MVP	0.0	Real number
+64	S.2PID	α	0.0	Real number
+66	S.2PID	β	1.0	
⋮			⋮	⋮
+90			0.0	Real number
+92			0.0	Real number
+94			0.0	Real number

The symbols in the ladder diagram mean the following.



Instruction name	S.IN	S.PHPL	S.2PID	S.OUT1
1) Input data head device	R0	R20	R40	R60
2) Block memory head device	R100	R120	R140	R160
3) Operation constant head device	R200	R220	R240	R260
4) Loop tag memory head device	R1000			
5) Set value head device	—	—	R300	—

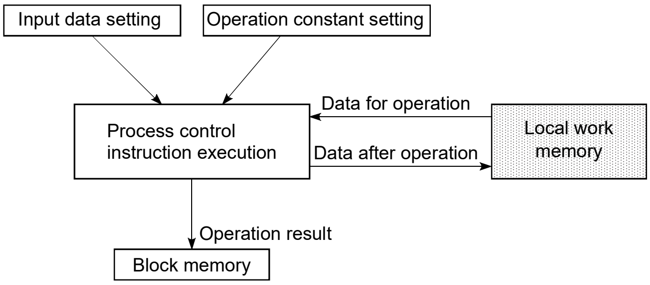
3.2 Local Work Memory

Local work memory is used as a temporary storage area in process control instruction operation. (The memory is used for each micro block.)

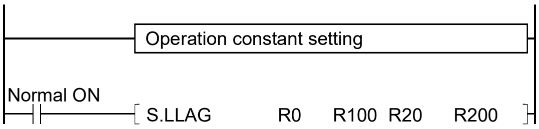
The following instructions use the local work memory.

Instruction	Remarks
S.LLAG (Lead-Lag) S.D (Derivative) S.DED (Dead time) S.FLT (Standard filter) S.BUMP (Bumpless transfer) S.AT1 (Auto tuning)	The system stores the midway operation results. (For system use only)
S.FG (Function generator) S.IFG (Inverse function generator)	The user stores the coordinate values (Xn, Yn) of a function generator. Operations are performed based on these values.

[Block diagram]



[Ladder diagram]



Instruction name	S.LLAG (Lead-Lag)
Input data head device	R0
Block memory head device	R100
Operation constant head device	R20
Local work memory head device	R200

The application of the local work memory changes depending on the used instruction. Refer to the explanation section of the corresponding instruction.

3.3 Data Used for Process Control Instructions

The following data are used for the process control instructions.

• Loop tag memory	Page 27, Section 3.3.1
• Input data	Page 28, Section 3.3.2
• Block memory	Page 29, Section 3.3.3
• Operation constant	Page 29, Section 3.3.4
• Local work memory	Page 26, Section 3.2

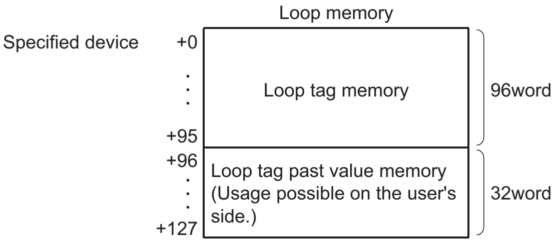
3.3.1 Loop memory

(1) Loop memory

(a) The loop memory is an area that stores the data used commonly by the process control instructions specified as the loop type.
The loop memory also has an area that stores the data used by the CPU module system during process control instruction execution.

(b) The loop memory has the "loop tag memory" and "loop tag past value memory" areas.

(c) The loop memory consists of 128 words (word device: 128 points).
When setting the loop memory areas, specify the device that can occupy 128 words consecutively.



(2) Loop tag memory

(a) The loop tag memory is an area that stores the data used commonly by the process control instructions specified as the loop type indicated in Page 22, Section 2.3.

(b) The loop tag memory consists of 96 words.

(c) Refer to Page 261, Appendix 2 (Loop tag memory list) for the applications of the area used by the process control instructions in the loop tag memory.

(3) Loop tag past value memory

(a) The loop tag past value memory is an area used by the CPU module system at the time of process control instruction execution.
The user cannot write data to this memory during run.
If the user writes data to the loop tag past value memory during run, normal operation cannot be performed.

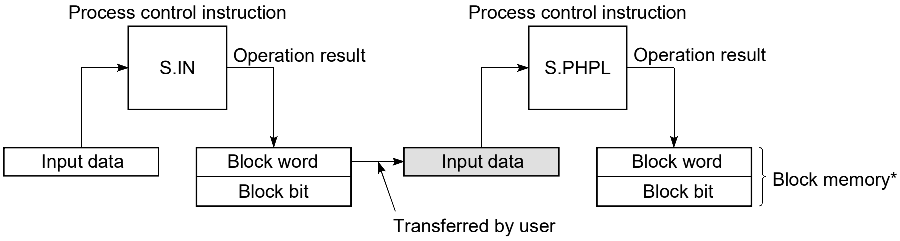
(b) The loop tag past value memory is a 32-word area after the loop tag memory.

(c) At the start of the process control instruction, write "0" to the loop tag past value memory.

3.3.2 Input data

(1) Input data is variable data given to each process control instruction.

(2) The input data uses the block word of the block memory that stores the operation result of the process control instruction executed previously.



*For the block memory, refer to Page 29, Section 3.3.3.

(3) The application of the input data changes depending on the used instruction. Refer to the explanation section of the corresponding instruction.

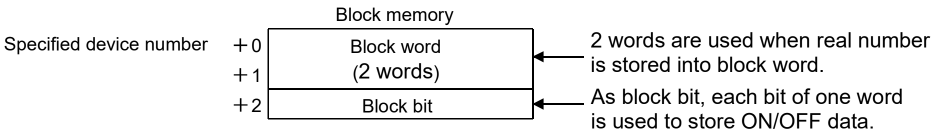
3.3.3 Block memory

The block memory is an area that stores the output information of the corresponding process control instruction.

The block memory has "block words" and "block bits".

The application of the block memory changes depending on the used instruction.

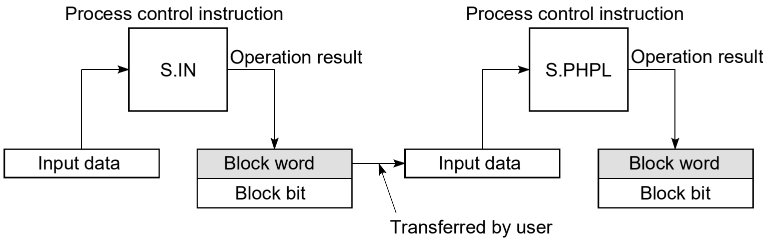
Refer to the explanation section of the corresponding instruction.


Block memory
Specified device number
+0
+1
+2
Block word
(2 words)
Block bit
2 words are used when real number is stored into block word.
As block bit, each bit of one word is used to store ON/OFF data.

(1) Block word (BW)

(a) The block word is an area that stores the operation result of the process control instruction.

(b) As the input data of the next process control instruction linked by the loop, the data stored in the block word is used.



(2) Block bit (BB)

The block bit is an area that stores the corresponding alarm data at process control instruction execution.

As the block bits, 16 bits of b0 to b15 are represented as BB1 to BB16.

The b0 (BB1) in the block bits stores "1" if any of b1 to b15 (BB2 to BB16) on each instruction stores an alarm data.

	b15			b12				b8				b4				b0
Block bit	BB16	BB15	BB14	BB13	BB12	BB11	BB10	BB9	BB8	BB7	BB6	BB5	BB4	BB3	BB2	BB1

3.3.4 Operation constant

(1) The operation constant is an area that stores the data used by only one process control instruction.

(2) The application of the operation constant changes depending on the used instruction. Refer to the explanation section of the corresponding instruction.

3.3.5 Loop tag memory allocation contents

The loop tag memory allocation contents are shown below.

Instructions used in loop tag

The number of words from the start of the loop tag

Abbreviated name of each item

Shaded values may change depending on the operation result.

For PID control (S2PID), the same values are used for all instructions in the same loop tag memory.

The offset values are fixed for each instruction

Instruction used	Offset	Item	Recommended range	Standard value setting	Data type
	+0				
	+1	MODE	0 to FFFF$_H$	8$_H$	BIN16bit
	+3	ALM	0 to FFFF$_H$	4000$_H$	BIN16bit
	+4	INH	0 to FFFF$_H$	0$_H$	BIN16bit
S.PHPL	+10	PV	RL to RH	0.0	Real number
S.OUT1	+12	MV	-10 to 110	0.0	Real number
S.2PID	+14	SV	RL to RH	0.0	Real number
S.2PID	+16	DV	-110 to 110	0.0	Real number
S.OUT1	+18	MH	-10 to 110	100.0	Real number
S.OUT1	+20	ML	-10 to 110	0.0	Real number
S.PHPL	+22	RH	-999999 to 999999	100.0	Real number
S.PHPL	+24	RL	-999999 to 999999	0.0	Real number
S.PHPL	+26	PH	RL to RH	100.0	Real number
S.PHPL	+28	PL	RL to RH	0.0	Real number
S.PHPL	+30	HH	RL to RH	100.0	Real number
S.PHPL	+32	LL	RL to RH	0.0	Real number
S.IN	+38	α	0 to 1	0.2	Real number
S.PHPL	+40	HS	0 to 999999	0.0	Real number
S.PHPL	+42	CTIM	0 to 999999	0.0	Real number
S.PHPL	+44	DPL	0 to 100	100.0	Real number
S.2PID	+46	CT	0 to 999999	1.0	Real number
S.OUT1	+48	DML	0 to 100	100.0	Real number
S.2PID	+50	DVL	0 to 100	100.0	Real number
S.2PID	+52	P	0 to 999999	1.0	Real number
S.2PID	+54	I	0 to 999999	10.0	Real number
S.2PID	+56	D	0 to 999999	0.0	Real number
S.2PID	+58	GW	0 to 100	0.0	Real number
S.2PID	+60	GG	0 to 999999	1.0	Real number
S.OUT1	+62	MVP	-999999 to 999999	0.0	Real number
S.2PID	+64	α	0 to 1	0.0	Real number
S.2PID	+66	β	0 to 1	1.0	

(1) This section describes the bit assignments of the following items.

(a) ALM

b15	b14	b13	b12	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

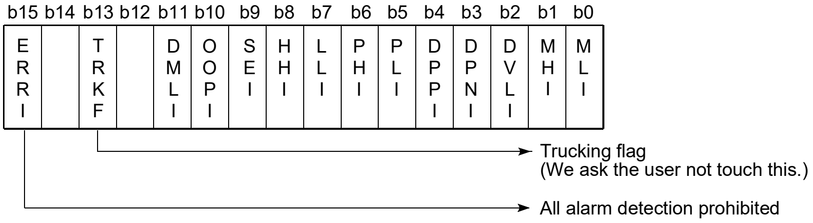
The default standard value, 4000$_H$, indicates manual operation in loop stop state. To enable auto alarm, set 0000$_H$.

S: Stored by the system
U: Set by the user

Name	Abbreviation	Description	Flag establishment conditions
Stop alarm	SPA	Shows the loop stop status. Changes the loop mode to manual. Conducts stop alarm processing for the output value (BW) and alarm signal.	U
Output change rate limit alarm	DMLA	Conducts the change rate limiter for the input data and outputs the change rate alarm. (For the output change upper limit value/control value).	S
Output open alarm	OOPA	Shows that it has changed to open status when the operation output signal has become disconnected, etc.	S
Sensor alarm	SEA	Sensor error alarm	S
Upper upper limit alarm	HHA	Checks the upper limit value of the process equipment upper limit, and outputs an alarm if the process value is higher than the upper limit value.	S
Lower lower limit alarm	LLA	Checks the lower limit value of the process equipment lower limit, and outputs an alarm if the process value is lower than the lower limit value.	S
Upper limit alarm	PHA	Checks the upper limit value of the process value, and outputs an alarm if the process value is higher than the upper limit value.	S
Lower limit alarm	PLA	Checks the lower limit value of the process value, and outputs an alarm if the process value is lower than the lower limit value.	S
Positive direction change rate alarm	DPPA	Outputs an alarm if the change rate is higher than the upward trend change rate range.	S
Negative direction change rate alarm	DPNA	Outputs an alarm if the change rate is lower than the downward trend change rate range.	S
Deviation large alarm	DVLA	Conducts an error check and then outputs an alarm if over. In addition, if the error check determines that the deviation is completely less than the warning value and the error is reduced by a set value from the warning value then the deviation large alarm will be released.	S
Output upper limit alarm	MHA	Conducts a check using the upper/lower limiter and if the limiter results are larger than the input upper limit value an alarm is output.	S
Output lower limit alarm	MLA	A check is conducted by an upper/lower limiter and if the limiter results are smaller than the input lower limit value an alarm is output.	S

(b) INH

This prohibits alarm detection for each item. In addition, the alarms prohibited by INH are not detected. (The INH bits 0 to 11 correspond to the bits 0 to 11 of ALM.)



b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ERRI		TRKF		DMLI	OOPI	SEI	HHI	LLI	PHI	PLI	DPPI	DPNI	DVLI	MHI	MLI

Trucking flag
(We ask the user not touch this.)
All alarm detection prohibited

(c) MODE

The process control instructions have the following operation modes that satisfy the following operations in a system connected to an operator station, programmable controller, host computer, machine side operation panel and like.

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
					CSV	CMV	CCB	CAB	CMB	CAS	AUT	MAN	LCC	LCA	LCM

For MODE make one of them a 1 bit only flag 1.

Operation mode	Description	Application
MAN (MANUAL)	• Manual operation from OPS • SV and MV can be set.	Monitoring and control from operator station are performed.
AUT (AUTOMATIC)	• Automatic operation • SV can be set. • MV cannot be set.	
CAS (CASCADE)	• Cascade operation • SV and MV cannot be set.	
CMV (COMPUTER MV)	• Automatic MV setting from host computer	Loop operation from host computer can be performed and operation mode is controlled and monitored at operator station.
CSV (COMPUTER SV)	• Automatic SV setting from host computer	
CMB (COMPUTER MANUAL BACK UP)	• Manual operation backup when host computer is abnormal	During loop control by host computer, backup is provided by predetermined operator station when computer fails.
CAB (COMPUTER AUTOMATIC BACK UP)	• Automatic operation backup when host computer is abnormal	
CCB (COMPUTER CASCADE BACK UP)	• Cascade operation backup when host computer is abnormal	
LCM (LOCAL MANIPULATED)	• Local manual operation	At startup of plant, operation and startup are performed by loop display or like from other than operator station and operation mode is monitored by operator station.
LCA (LOCAL AUTOMATIC)	• Local automatic operation	
LCC (LOCAL CASCADE)	• Local cascade operation	

CHAPTER 4 HOW TO EXECUTE PROCESS CONTROL INSTRUCTIONS

4.1 Execution Cycle and Control Cycle

(1) Execution cycle

(a) An execution cycle is an interval at which the process control instruction is executed.

(b) There are the following methods to execute the process control instruction in each execution cycle.

1) Method using timer
A timer is used to measure the execution cycle and the process control instruction is executed when the timer times out.

2) Method using interrupt programs
Any of interrupt programs of I28 to I31 is run in each execution cycle.

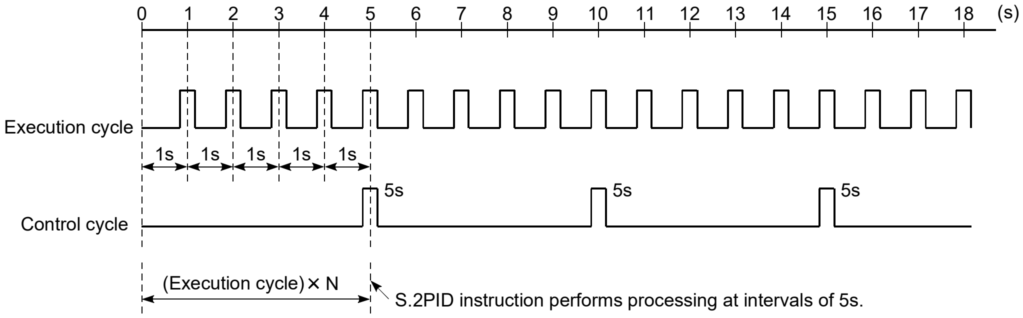
3) Method using fixed scan execution type program
A fixed scan execution type program is run in each execution cycle.

(c) Specify in the special registers (SD1500, SD1501) the value of the execution cycle used for the process control instruction as a real number.

(2) Control cycle

(a) A control cycle is an interval in which PID control is performed for an instruction such as S.2PID (2-degree-of-freedom PID).
As the control cycle, specify an integral multiple of the execution cycle.
The S.2PID or similar instruction counts the execution cycle in each execution cycle and starts PID operation when the specified control cycle is reached.

(b) Specify in the loop tag memory (See Page 27, Section 3.3.1) the control cycle used for the S.2PID or similar instruction.
The S.2PID or similar instruction uses the value of the control cycle specified in the loop tag memory to perform PID control.
Example) When monitoring is performed at intervals of 1s in 2-degree-of-freedom PID control and PID control is carried out at intervals of 5s.

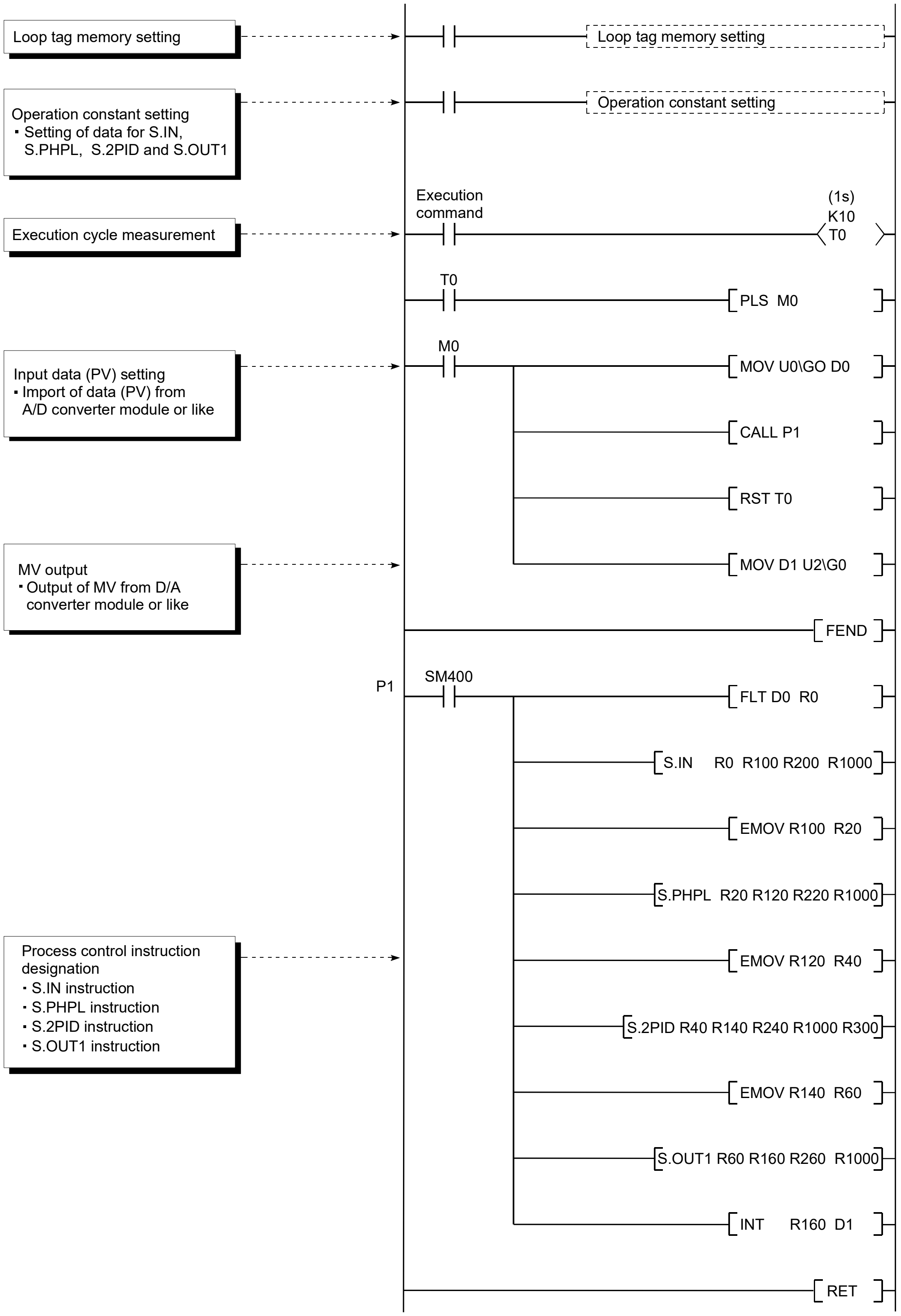

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 (s)
Execution cycle
1s
1s
1s
1s
1s
5s
5s
5s
Control cycle
(Execution cycle) × N
S.2PID instruction performs processing at intervals of 5s.

Point

When the control cycle is set to an integral multiple of the execution cycle, monitoring such as a PV check can be performed in each execution cycle.

4.2 Concept of Program

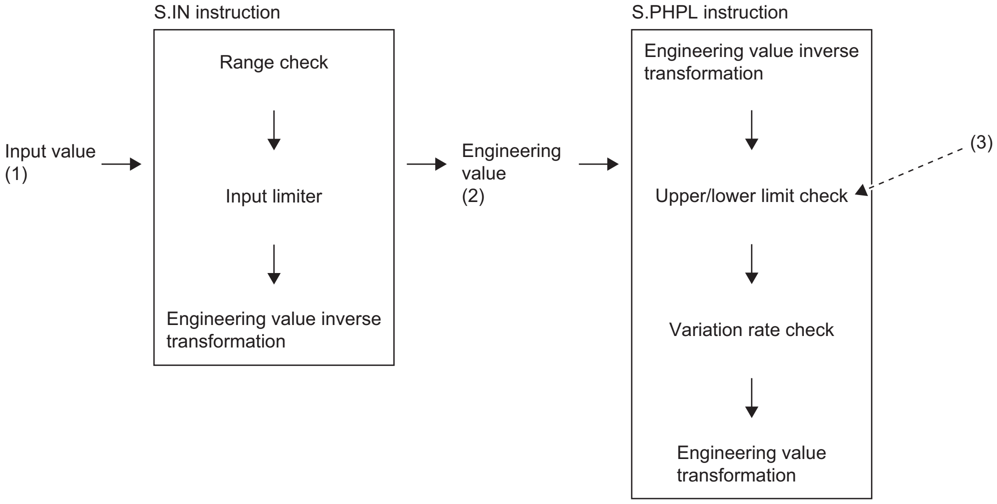
[Program example using S.2PID instruction at execution cycle of 1s]


Loop tag memory setting
Loop tag memory setting
Operation constant setting
・Setting of data for S.IN, S.PHPL, S.2PID and S.OUT1
Operation constant setting
Execution cycle measurement
Execution command
(1s)
K10
T0
T0
PLS M0
Input data (PV) setting
・Import of data (PV) from A/D converter module or like
M0
MOV U0\GO D0
CALL P1
RST T0
MV output
・Output of MV from D/A converter module or like
MOV D1 U2\G0
FEND
P1
SM400
FLT D0 R0
S.IN R0 R100 R200 R1000
EMOV R100 R20
S.PHPL R20 R120 R220 R1000
Process control instruction designation
・S.IN instruction
・S.PHPL instruction
・S.2PID instruction
・S.OUT1 instruction
EMOV R120 R40
S.2PID R40 R140 R240 R1000 R300
EMOV R140 R60
S.OUT1 R60 R160 R260 R1000
INT R160 D1
RET

4.2.1 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.

Example) 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 to 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.

Example 1) Adding a program to disable alarm detection

Add a program that performs the following processing.

(a) When RH = HH, the bit HHI of the disable alarm detection (INH) turns on.

(b) When RH = PH, the bit PHI of the disable alarm detection (INH) turns on.

(c) When RL = LL, the bit LLI of the disable alarm detection (INH) turns on.

(d) When RL = PL, the bit PLI of the disable alarm detection (INH) turns on.

Example 2) 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)

CHAPTER 5 EXECUTION CONDITION SWITCHING AND FUNCTIONS

5.1 Execution Condition Switching

5.1.1 Loop RUN/STOP

If any loop component such as a detector or operation end other than the programmable controller fails, each loop can be run/stopped to perform the maintenance of the corresponding loop.

The "SPA" bit of the alarm detection (ALM) is used to run/stop the corresponding loop.

(1) Basic operation during loop STOP

(a) Output status hold (The S.2PID instruction is output = 0)

(b) Alarm No detection (Process alarm)

(c) Make the control mode MAN.

5.2 Functions

5.2.1 Tracking function

The tracking function includes the "bumpless function" and "output limiter processing".

(1) Bumpless function
The bumpless function prevents manipulated value (MV) output stepping changes when switching from the automatic mode to manual mode and continuously controls MV output.

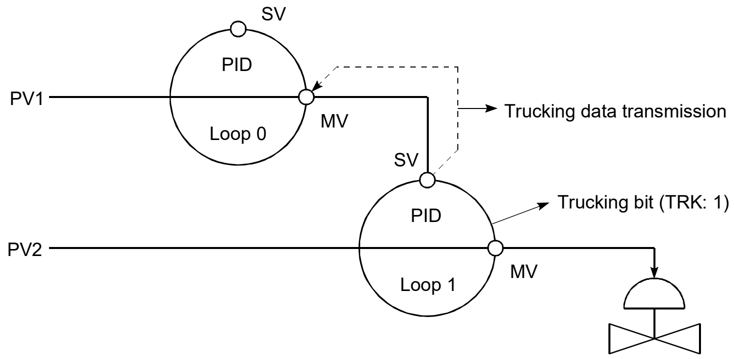
(2) Output limiter processing function
The output limiter processing function limits the upper limit and lower limit of the manipulated value (MV) output by the PID operation during the automatic mode. This output limiter processing function is only valid in the automatic mode and is not executed for manual data. In addition, when the parameter tracking function execution validity is set to not valid when in the automatic mode the output limiter processing function will not execute.

5.2.2 Cascade loop tracking

The process control loops that comprise 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 the sudden variation of the set value (SV) when the operation mode of the secondary loop (Loop 1) is changed.

(1) The cascade PID loop Tracking processing is shown in the diagram below.
[Processing concept diagram]



(a) In cascade operation, the manipulated value (MV) of Loop 0 is transferred to the set value (SV) of Loop 1.

(b) 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)

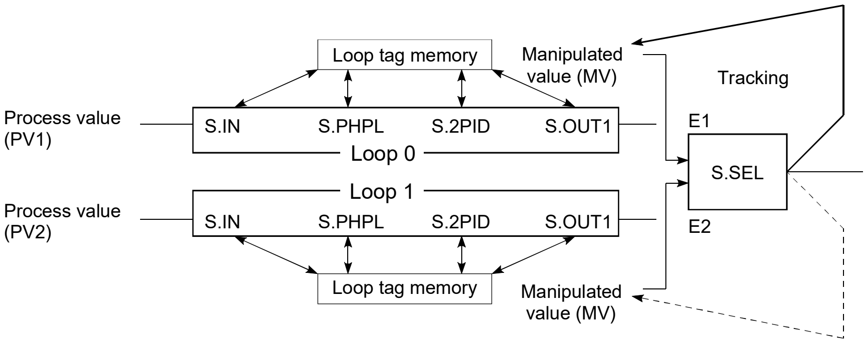
(2) Make the following settings to perform tracking.
(Tracking is performed when the operation mode is switched to other than CAS, CSV or CCB.)
For 2-degree-of-freedom PID (S.2PID), set the following operation constant items to specify tracking.

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 used)

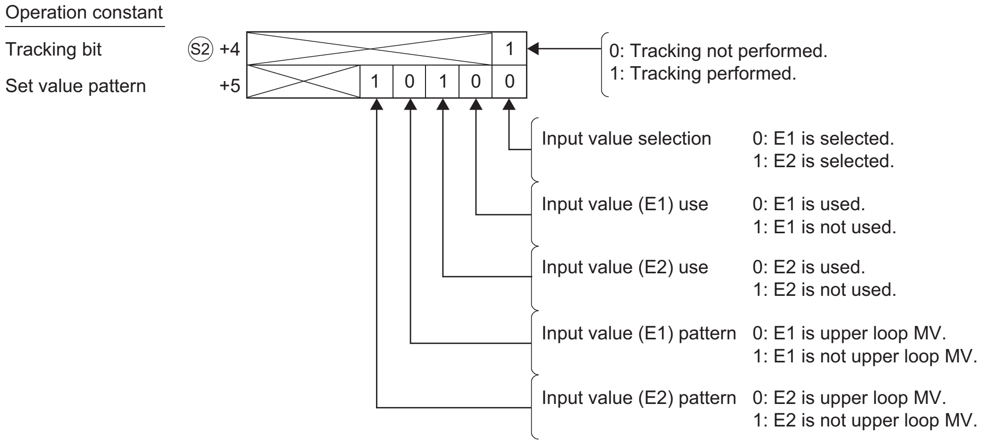
5.2.3 Loop selector tracking

Tracking is performed under the following conditions.

- The operation mode is any of MAN, CMB, CMV and LCM and the tracking bit (TRK) is 1
- When the operation mode is any of AUT, CAS, CAB, CCB, CSV, LCA and LCC
  The tracking bit (TRK) is 1 and BB1 of BB is 1



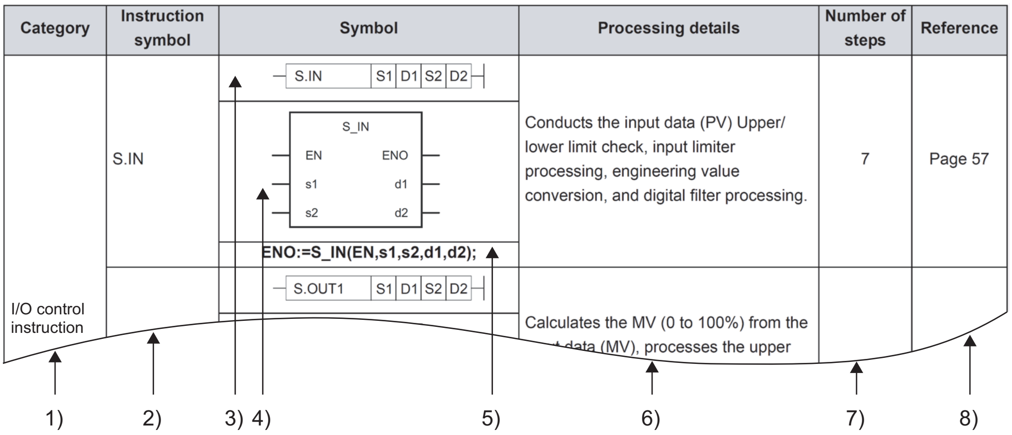
Example) When the S.SEL instruction uses the input value E1 and E1 uses the upper loop (loop 0) MV, the S.SEL instruction's MV is trucked to loop 0's MV.
The setting that conducts Tracking is shown below.



CHAPTER 6 INSTRUCTIONS

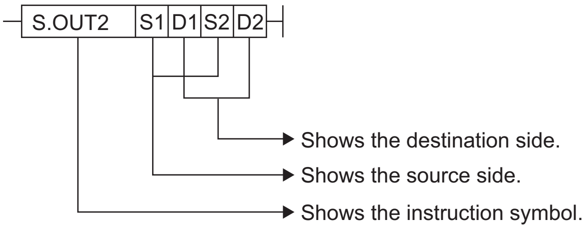
6.1 How to Read the Instruction List

Process control instructions are classified into six categories: I/O control instructions, control operation instructions, compensation operation instructions, arithmetic operation instructions, comparison operation instructions, and auto tuning instructions.



Category	Instruction symbol	Symbol	Processing details	Number of steps	Reference
I/O control instruction	S.IN	S.IN S1 D1 S2 D2 S_IN EN ENO s1 d1 s2 d2 ENO:=S_IN(EN,s1,s2,d1,d2);	Conducts the input data (PV) Upper/ lower limit check, input limiter processing, engineering value conversion, and digital filter processing.	7	Page 57
		S.OUT1 S1 D1 S2 D2	Calculates the MV (0 to 100%) from the ... data (MV), processes the upper		

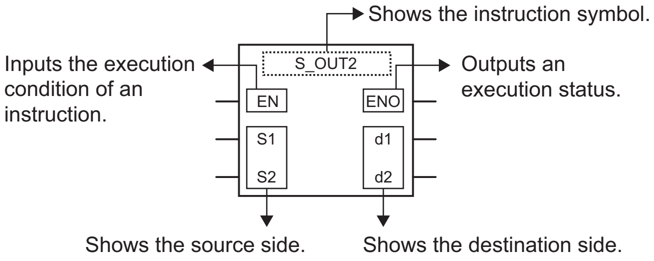
1) Instructions are classified by their application.
2) An instruction symbol used in the program
3) A written format in the ladder diagram



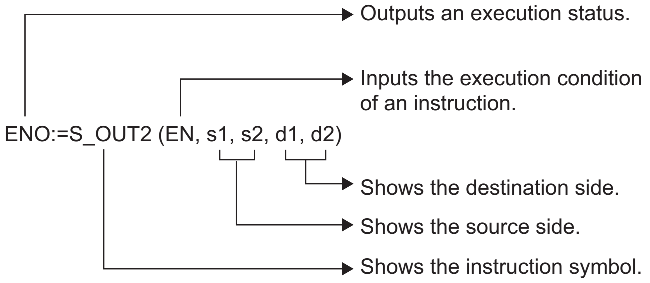
Destination: Shows the destination of the data after operation.

Source: Stores the data before the operation.

4) A written format in the structured ladder/FBD


Shows the instruction symbol.
Inputs the execution condition of an instruction.
S_OUT2
Outputs an execution status.
EN
ENO
S1
d1
S2
d2
Shows the source side.
Shows the destination side.

5) A written format in the structured text language



```
ENO:=S_OUT2 (EN, s1, s2, d1, d2)
```

Outputs an execution status.
Inputs the execution condition of an instruction.
Shows the destination side.
Shows the source side.
Shows the instruction symbol.

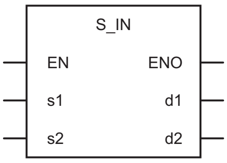
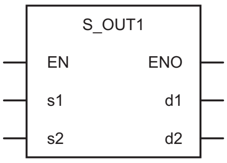
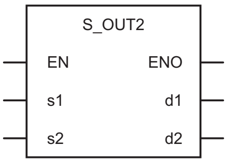
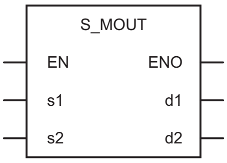
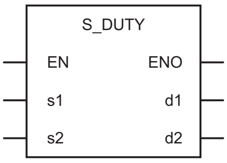
6) Details of processing performed by the instruction

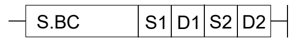
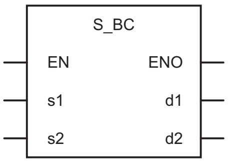
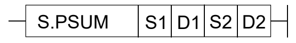
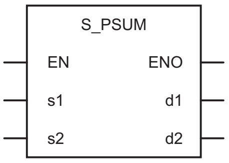
7) The number of steps in the instruction. For details, refer to Page 21, Section 2.2.7.

8) Pages to be referred to

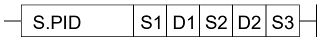
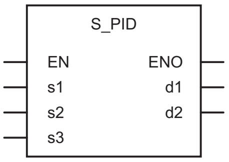
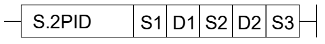
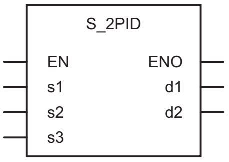
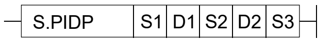
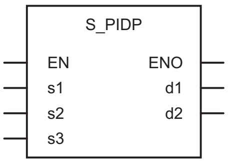
6.2 List of Instructions

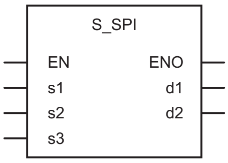
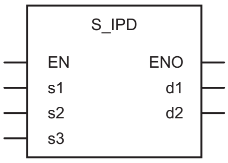
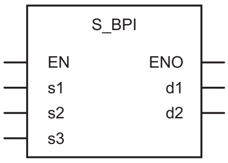
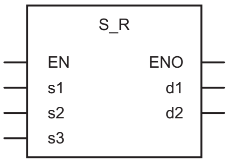
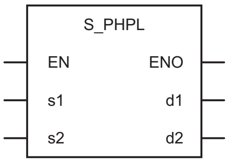
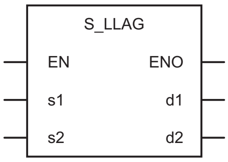
6.2.1 I/O control instructions

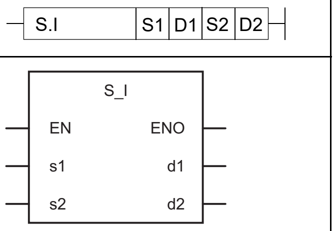
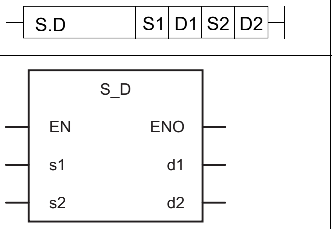
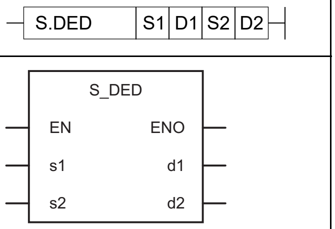
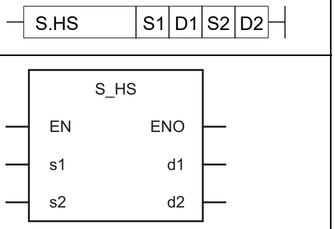
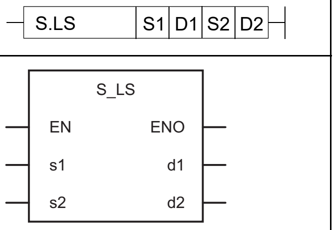
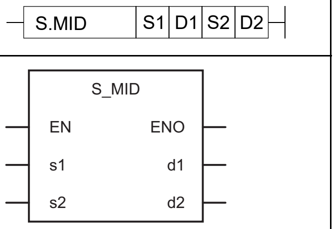
Category	Instruction symbol	Symbol	Processing details	Number of steps	Reference
I/O control instruction	S.IN	S.IN S1 D1 S2 D2  **ENO:=S_IN(EN,s1,s2,d1,d2);**	Conducts the input data (PV) Upper/ lower limit check, input limiter processing, engineering value conversion, and digital filter processing.	7	Page 55
	S.OUT1	S.OUT1 S1 D1 S2 D2  **ENO:=S_OUT1(EN,s1,s2,d1,d2);**	Calculates the MV (0 to 100%) from the input data (ΔMV), performs change rate, upper/lower limiter processing and output on time conversion.	8	Page 60
	S.OUT2	S.OUT2 S1 D1 S2 D2  **ENO:=S_OUT2(EN,s1,s2,d1,d2);**	Performs change rate, upper/lower limiter processing and output on time conversion from the input data (MV).	8	Page 66
	S.MOUT	S.MOUT S1 D1 S2 D2  **ENO:=S_MOUT(EN,s1,s2,d1,d2);**	Reads the manipulated value (MV) of the loop tag memory and performs output conversion.	8	Page 71
	S.DUTY	S.DUTY S1 D1 S2 D2  **ENO:=S_DUTY(EN,s1,s2,d1,d2);**	Changes the ON/OFF rate within a given cycle in proportion to the input data (0 to 100%) and outputs the result.	8	Page 75

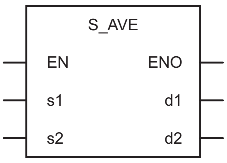
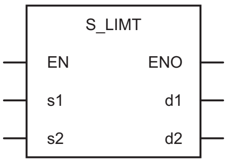
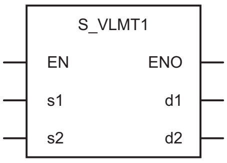
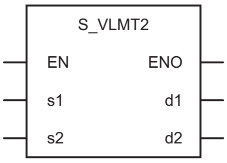
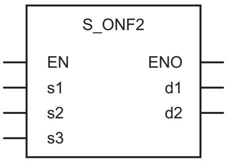
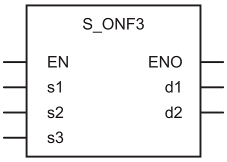
Category	Instruction symbol	Symbol	Processing details	Number of steps	Reference
I/O control instruction	S.BC	  **ENO:=S_BC(EN,s1,s2,d1,d2);**	Compares the input data with the set value and outputs bit data as soon as the input data reaches the set value.	7	Page 82
	S.PSUM	  **ENO:=S_PSUM(EN,s1,s2,d1,d2);**	Integrates the number of input pulses and outputs the result.	8	Page 86

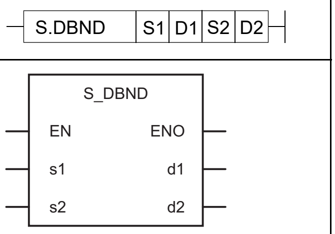
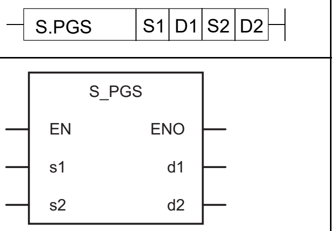
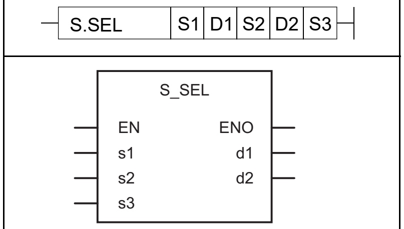
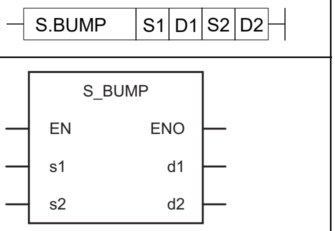
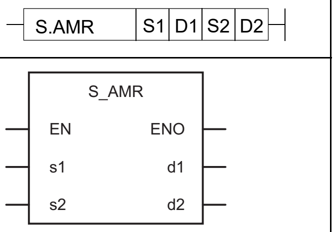
6.2.2 Control operation instructions

Category	Instruction symbol	Symbol	Processing details	Number of steps	Reference
Control operation instruction	S.PID	  **ENO:=S_PID(EN,s1,s2,s3,d1,d2);**	Conducts process value derivative type PID operations. (Incomplete derivative) Performs SV setting processing, tracking processing, gain Kp operation processing, PID operation and deviation check.	9	Page 91
	S.2PID	  **ENO:=S_2PID(EN,s1,s2,s3,d1,d2);**	Performs 2-degree-of-freedom PID operation (incomplete derivative). Performs SV setting processing, tracking processing, gain Kp operation processing, 2-degree-of-freedom PID operation and deviation check.	9	Page 99
	S.PIDP	  **ENO:=S_PIDP(EN,s1,s2,s3,d1,d2);**	Performs position type PID operation. Performs SV setting processing, tracking processing, gain Kp operation processing, PID operation, deviation check and operation mode judgment. According to the result, performs change rate, upper/lower limiter and output on time conversion or performs alarm clear processing and output on time conversion.	9	Page 107

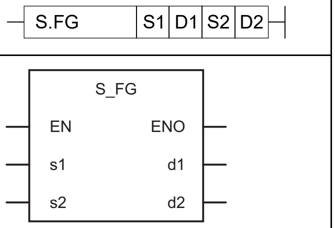
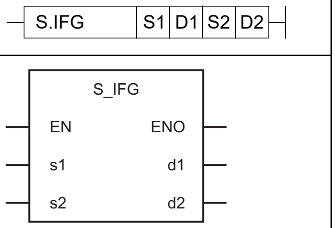
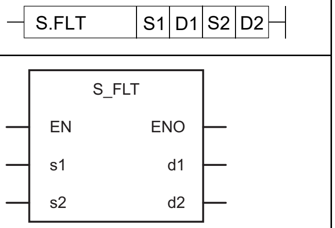
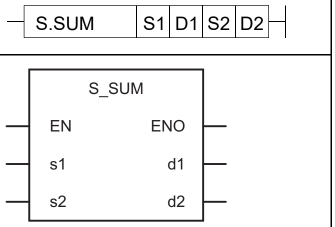
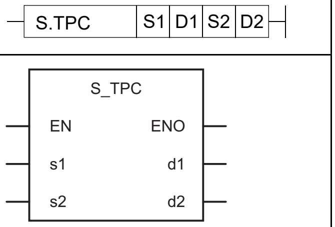
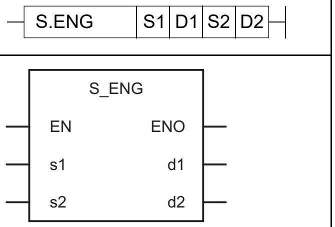
Category	Instruction symbol	Symbol	Processing details	Number of steps	Reference
Control operation instruction	S.SPI	S.SPI S1 D1 S2 D2 S3  **ENO:=S_SPI(EN,s1,s2,s3,d1,d2);**	Judges between the operating time and hold time, and if it is the operating time, performs SV setting processing, tracking processing, gain Kp operation processing, SPI operation and deviation check.	9	Page 116
	S.IPD	S.IPD S1 D1 S2 D2 S3  **ENO:=S_IPD(EN,s1,s2,s3,d1,d2);**	Performs I-PD operation. Performs SV setting processing, tracking processing, gain Kp operation processing, IPD operation and deviation check.	9	Page 123
	S.BPI	S.BPI S1 D1 S2 D2 S3  **ENO:=S_BPI(EN,s1,s2,s3,d1,d2);**	Performs blend PI operation. Performs SV setting processing, tracking processing, gain Kp operation processing, BPI operation and deviation check.	9	Page 131
	S.R	S.R S1 D1 S2 D2 S3  **ENO:=S_R(EN,s1,s2,s3,d1,d2);**	Performs engineering value conversion, tracking processing, change rate limiter and ratio operation on the input data.	8	Page 138
	S.PHPL	S.PHPL S1 D1 S2 D2  **ENO:=S_PHPL(EN,s1,s2,d1,d2);**	Conducts an Upper limit value/lower limit value check of the PV output by the S.IN instruction.	8	Page 143
	S.LLAG	S.LLAG S1 D1 S2 D2  **ENO:=S_LLAG(EN,s1,s2,d1,d2);**	Conducts lead-lag compensation for input data and outputs the operation results.	8	Page 149

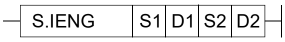
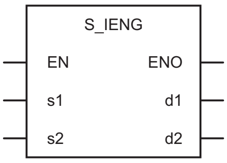
Category	Instruction symbol	Symbol	Processing details	Number of steps	Reference
Control operation instruction	S.I	 S.I S1 D1 S2 D2 S_I EN ENO s1 d1 s2 d2 **ENO:=S_I(EN,s1,s2,d1,d2);**	Conducts integral operations on the input data and outputs the operation results.	7	Page 151
	S.D	 S.D S1 D1 S2 D2 S_D EN ENO s1 d1 s2 d2 **ENO:=S_D(EN,s1,s2,d1,d2);**	Conducts Derivative operations on the input data and outputs the operation results.	7	Page 153
	S.DED	 S.DED S1 D1 S2 D2 S_DED EN ENO s1 d1 s2 d2 **ENO:=S_DED(EN,s1,s2,d1,d2);**	Delays the input data by the specified dead time and then outputs it.	8	Page 155
	S.HS	 S.HS S1 D1 S2 D2 S_HS EN ENO s1 d1 s2 d2 **ENO:=S_HS(EN,s1,s2,d1,d2);**	Outputs the maximum value among the input data.	7	Page 158
	S.LS	 S.LS S1 D1 S2 D2 S_LS EN ENO s1 d1 s2 d2 **ENO:=S_LS(EN,s1,s2,d1,d2);**	Outputs the minimum value among the input data.	7	Page 160
	S.MID	 S.MID S1 D1 S2 D2 S_MID EN ENO s1 d1 s2 d2 **ENO:=S_MID(EN,s1,s2,d1,d2);**	Outputs the middle value between the maximum value and minimum value among the input data.	8	Page 162

Category	Instruction symbol	Symbol	Processing details	Number of steps	Reference
Control operation instruction	S.AVE	S.AVE S1 D1 S2 D2  **ENO:=S_AVE(EN,s1,s2,d1,d2);**	Calculates and outputs the average value of the input data.	8	Page 165
	S.LIMT	S.LIMT S1 D1 S2 D2  **ENO:=S_LIMT(EN,s1,s2,d1,d2);**	Limits the output value with hysteresis.	8	Page 167
	S.VLMT1	S.VLMT1 S1 D1 S2 D2  **ENO:=S_VLMT1(EN,s1,s2,d1,d2);**	Limits the varying speed of the output value.	9	Page 169
	S.VLMT2	S.VLMT2 S1 D1 S2 D2  **ENO:=S_VLMT2(EN,s1,s2,d1,d2);**	Limits the varying speed of the output value.	9	Page 172
	S.ONF2	S.ONF2 S1 D1 S2 D2 S3  **ENO:=S_ONF2(EN,s1,s2,s3,d1,d2);**	Performs two-position ON/OFF control. Performs SV setting processing, tracking processing, MV compensation, MV output and two-position ON/OFF control.	9	Page 175
	S.ONF3	S.ONF3 S1 D1 S2 D2 S3  **ENO:=S_ONF3(EN,s1,s2,s3,d1,d2);**	Performs three-position ON/OFF control. Performs SV setting processing, tracking processing, MV compensation, MV output and three-position ON/OFF control.	9	Page 181

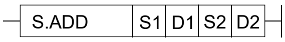
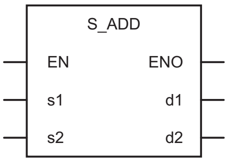
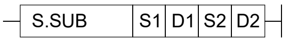
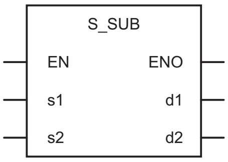
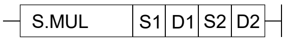
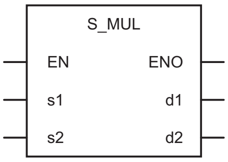
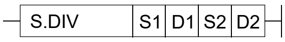
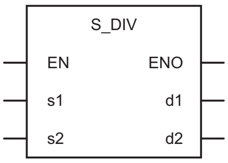
Category	Instruction symbol	Symbol	Processing details	Number of steps	Reference
Control operation instruction	S.DBND	 **ENO:=S_DBND(EN,s1,s2,d1,d2);**	Provides a dead band and performs output processing.	8	Page 187
	S.PGS	 **ENO:=S_PGS(EN,s1,s2,d1,d2);**	Provides a control output according to the SV and MV pattern.	8	Page 189
	S.SEL	 S.SEL S1 D1 S2 D2 S3 S_SEL EN ENO s1 d1 s2 d2 s3 **ENO:=S_SEL(EN,s1,s2,s3,d1,d2);**	Outputs the value selected by the selection signal out of the input data in the automatic mode, or outputs the MV of the loop tag memory in the manual mode.	9	Page 194
	S.BUMP	 **ENO:=S_BUMP(EN,s1,s2,d1,d2);**	Brings the output value closer to the output set value from the output control value gradually when the mode select signal is switched from manual to automatic.	8	Page 200
	S.AMR	 **ENO:=S_AMR(EN,s1,s2,d1,d2);**	Increases or decreases the output value at the fixed rate.	8	Page 203

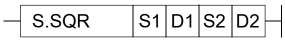
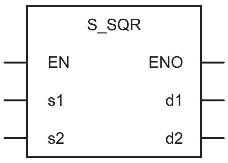
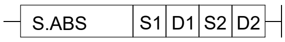
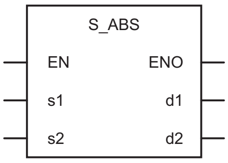
6.2.3 Compensation operation instructions

Category	Instruction symbol	Symbol	Processing details	Number of steps	Reference
Compensa-tion operation instruction	S.FG	 **ENO:=S_FG(EN,s1,s2,d1,d2);**	Outputs the value that follows the function generator pattern whose input data is specified.	7	Page 206
	S.IFG	 **ENO:=S_IFG(EN,s1,s2,d1,d2);**	Outputs the value that follows the inverse function generator pattern whose input data is specified.	8	Page 208
	S.FLT	 **ENO:=S_FLT(EN,s1,s2,d1,d2);**	Outputs the average value of n pieces of data sampled at the specified data collection intervals.	8	Page 210
	S.SUM	 **ENO:=S_SUM(EN,s1,s2,d1,d2);**	Integrates the input data and outputs the result.	8	Page 213
	S.TPC	 **ENO:=S_TPC(EN,s1,s2,d1,d2);**	Makes temperature/pressure correction to the input data and outputs the result.	8	Page 215
	S.ENG	 **ENO:=S_ENG(EN,s1,s2,d1,d2);**	Converts the input data into an engineering value.	8	Page 218

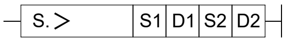
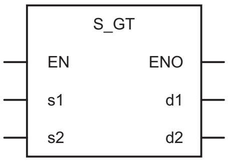
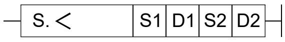
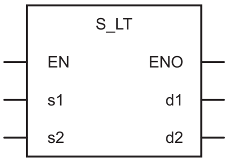
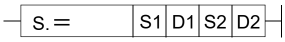
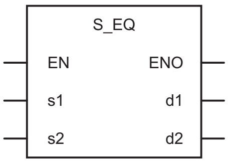
Category	Instruction symbol	Symbol	Processing details	Number of steps	Reference
Compensation operation instruction	S.IENG	  **ENO:=S_IENG(EN,s1,s2,d1,d2);**	Reversely converts the input data from the engineering value and outputs the result.	8	Page 220

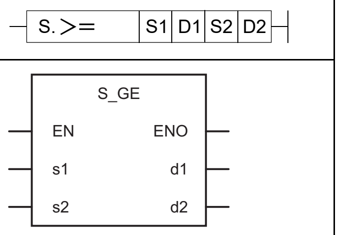
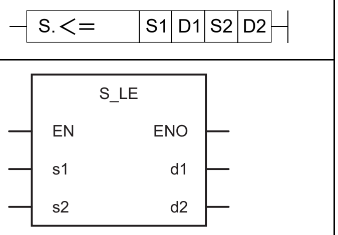
6.2.4 Arithmetic operation instructions

Category	Instruction symbol	Symbol	Processing details	Number of steps	Reference
Arithmetic operation instruction	S.ADD	  **ENO:=S_ADD(EN,s1,s2,d1,d2);**	Adds the input data with coefficients.	8	Page 222
	S.SUB	  **ENO:=S_SUB(EN,s1,s2,d1,d2);**	Subtracts the input data with coefficients.	8	Page 224
	S.MUL	  **ENO:=S_MUL(EN,s1,s2,d1,d2);**	Multiplies the input data with coefficients.	8	Page 226
	S.DIV	  **ENO:=S_DIV(EN,s1,s2,d1,d2);**	Divides the input data with coefficients.	8	Page 228

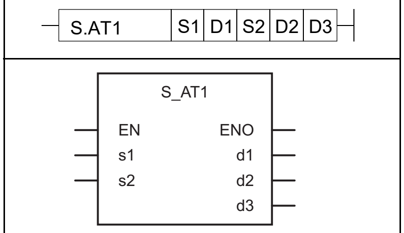
Category	Instruction symbol	Symbol	Processing details	Number of steps	Reference
Arithmetic operation instruction	S.SQR	  **ENO:=S_SQR(EN,s1,s2,d1,d2);**	Outputs the square root (√) of the input data.	8	Page 230
	S.ABS	  **ENO:=S_ABS(EN,s1,s2,d1,d2);**	Outputs the absolute value of the input data.	8	Page 232

6.2.5 Comparison operation instructions

Category	Instruction symbol	Symbol	Processing details	Number of steps	Reference
Comparison operation instruction	S. >	  **ENO:=S_GT(EN,s1,s2,d1,d2);**	Compares the input data and outputs the result of comparison.	7	Page 234
	S. <	  **ENO:=S_LT(EN,s1,s2,d1,d2);**	Compares the input data and outputs the result of comparison.	7	Page 236
	S. =	  **ENO:=S_EQ(EN,s1,s2,d1,d2);**	Compares the input data and outputs the result of comparison.	7	Page 238

Category	Instruction symbol	Symbol	Processing details	Number of steps	Reference
Comparison operation instruction	S. >=	 S.>= S1 D1 S2 D2 S_GE EN ENO s1 d1 s2 d2 **ENO:=S_GE(EN,s1,s2,d1,d2);**	Compares the input data and outputs the result of comparison.	7	Page 240
	S. <=	 S.<= S1 D1 S2 D2 S_LE EN ENO s1 d1 s2 d2 **ENO:=S_LE(EN,s1,s2,d1,d2);**	Compares the input data and outputs the result of comparison.	7	Page 242

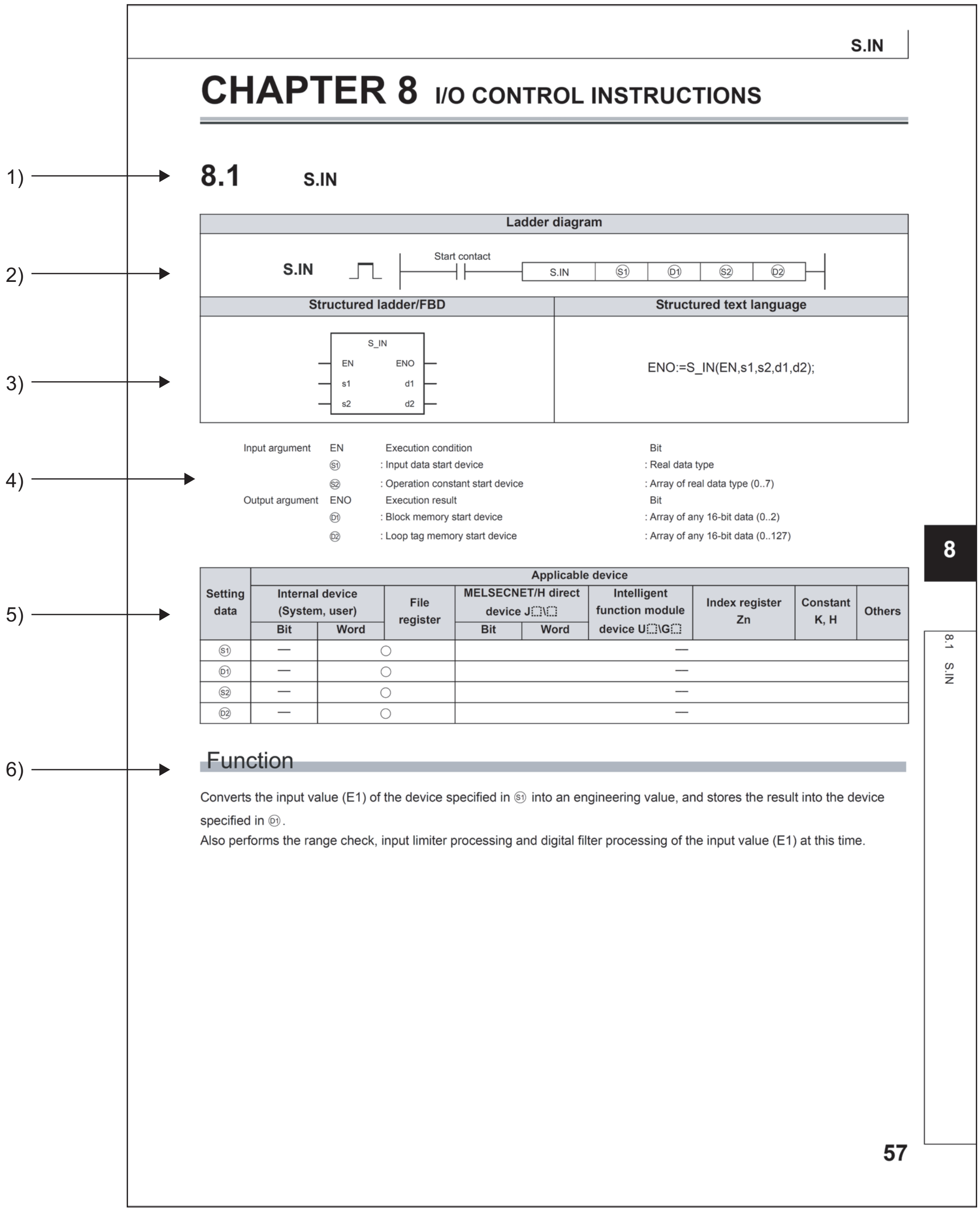
6.2.6 Auto tuning instructions

Category	Instruction symbol	Symbol	Processing details	Number of steps	Reference
Auto Tuning Instruction	S.AT1	 S.AT1 S1 D1 S2 D2 D3 S_AT1 EN ENO s1 d1 s2 d2 d3 **ENO:=S_AT1(EN,s1,s2,d1,d2,d3);**	Performs auto tuning and makes the initial setting of the PID constants.	9	Page 247

CHAPTER 7 HOW TO READ INSTRUCTION DETAILS

This chapter explains the page layout for chapters that describe instruction details (Chapter 8 to 13).

The descriptions in this chapter are for explanation purpose only, and are different from the actual pages.



S.IN

CHAPTER 8 I/O CONTROL INSTRUCTIONS

8.1 S.IN

Ladder diagram

S.IN

Structured ladder/FBD

Structured text language

ENO:=S_IN(EN,s1,s2,d1,d2);

Input argument	EN	Execution condition	Bit
	S1	: Input data start device	: Real data type
	S2	: Operation constant start device	: Array of real data type (0..7)
Output argument	ENO	Execution result	Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Loop tag memory start device	: Array of any 16-bit data (0..127)

Setting data	Applicable device								
	Internal device (System, user)		File register	MELSECNET/H direct device J□\□		Intelligent function module device U□\G□	Index register Zn	Constant K, H	Others
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

Function

Converts the input value (E1) of the device specified in S1 into an engineering value, and stores the result into the device specified in D1.

Also performs the range check, input limiter processing and digital filter processing of the input value (E1) at this time.

8

8.1 S.IN

57

1) A section number, instruction name, and instruction symbol

2) A written format in the ladder diagram and execution conditions

Execution condition	During on
Symbol shown on the reference page	⎍

3) Written formats in the structured ladder/FBD and structured text language

4) Explanations of the setting data. In Structured projects, replace the setting data Ⓢ1 and Ⓓ1 with s1 and d1.
For details on data types, refer to the MELSEC-Q/L/F Structured Programming Manual (Fundamentals).

5) A list of available devices. Devices with ○ are available.
Devices are classified as shown below.

Setting data	Internal devices (System, User)		File register R, ZR	Link direct device [3] J□\□		Intelligent function module U□\G□	Index register Zn	Constant[4]	Others[4]
	Bit	Word		Bit	Word				
Applicable devices[1]	X, Y, M, L, SM, F, B, SB, FX, FY[2]	T, ST, C, D, W, SD, SW, FD, @□	R, ZR	J□\X J□\Y J□\B J□\SB	J□\W J□\SW	U□\G□	Z	K, H , E, $	P, I, J, U, DX, DY, N, BL, TR, BL \ S,V

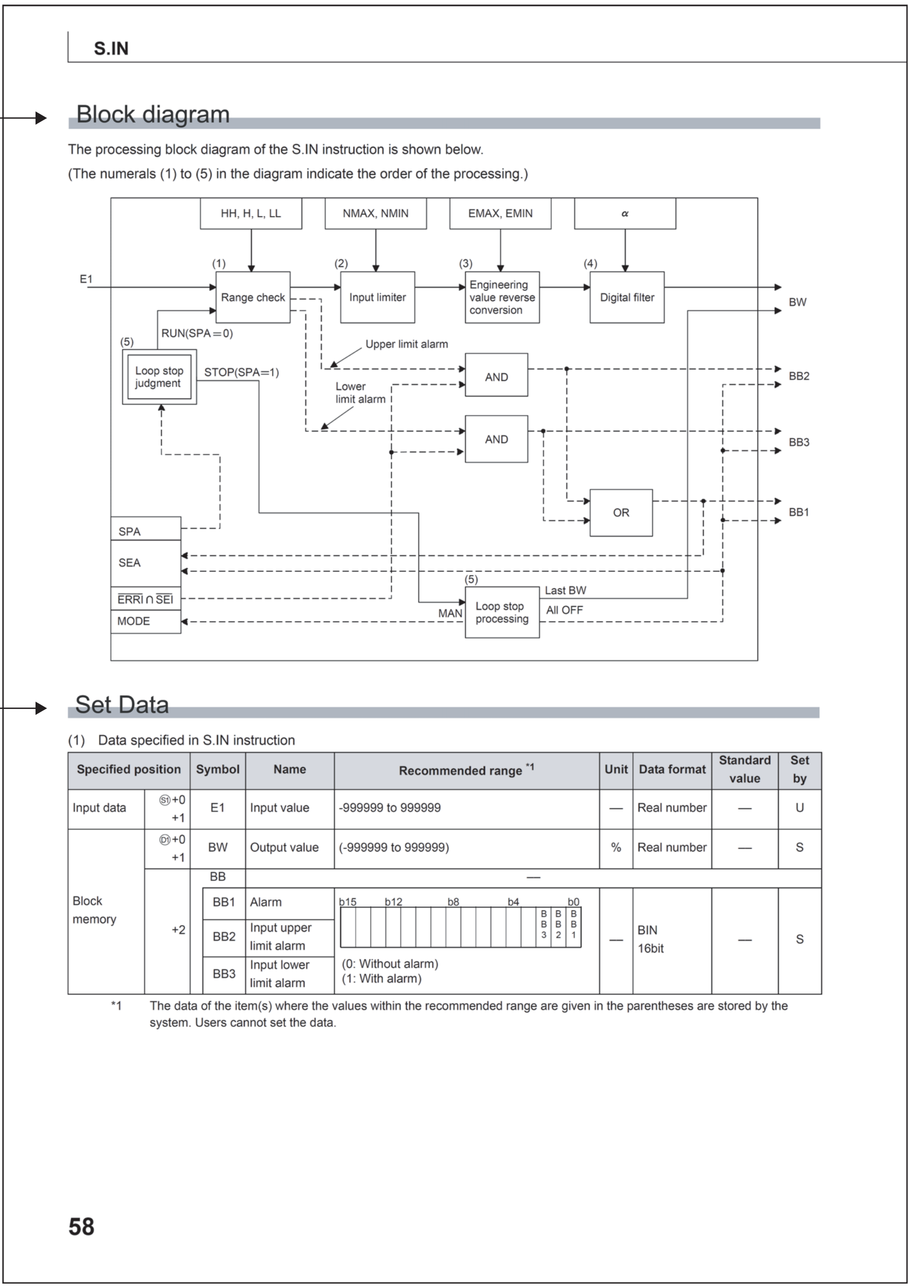
*1: For the description for the individual devices, refer to the User's Manual (Function Explanation, Program Fundamentals) for the CPU module used.

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

*3: Usable with the CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, and MELSECNET/10

*4: Devices which can be set are recorded in the "Constant" and the "Other" columns.

6) The function of the instruction



S.IN

7)

Block diagram

The processing block diagram of the S.IN instruction is shown below.

(The numerals (1) to (5) in the diagram indicate the order of the processing.)

8)

Set Data

(1) Data specified in S.IN instruction

Specified position		Symbol	Name	Recommended range *1	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value	-999999 to 999999	—	Real number	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	%	Real number	—	S
	+2	BB	—					
		BB1	Alarm	b15 b12 b8 b4 b0 BB3 BB2 BB1 (0: Without alarm) (1: With alarm)	—	BIN 16bit	—	S
		BB2	Input upper limit alarm					
		BB3	Input lower limit alarm					

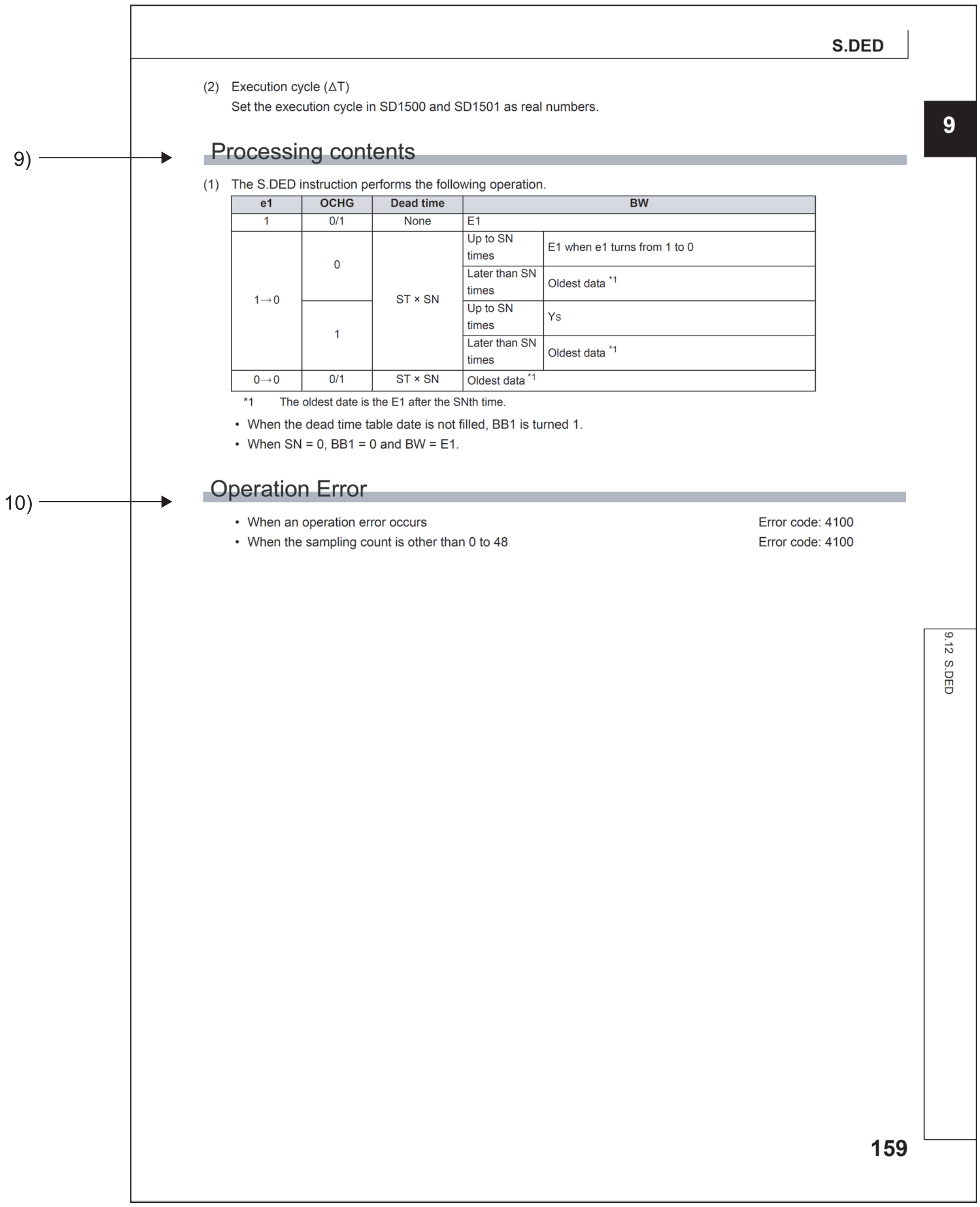
*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

58

7) The processing flow of the instruction

8) List of control data. S and U in the Set by column means:

- S: Stored by the system
- U: Set by the user



S.DED

9

(2) Execution cycle (ΔT)
Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) The S.DED instruction performs the following operation.

e1	OCHG	Dead time	BW	
1	0/1	None	E1	
1→0	0	ST × SN	Up to SN times	E1 when e1 turns from 1 to 0
			Later than SN times	Oldest data *1
	1		Up to SN times	Ys
			Later than SN times	Oldest data *1
0→0	0/1	ST × SN	Oldest data *1	

*1 The oldest date is the E1 after the SNth time.

- When the dead time table date is not filled, BB1 is turned 1.
- When SN = 0, BB1 = 0 and BW = E1.

Operation Error

- When an operation error occurs Error code: 4100
- When the sampling count is other than 0 to 48 Error code: 4100

9.12 S.DED

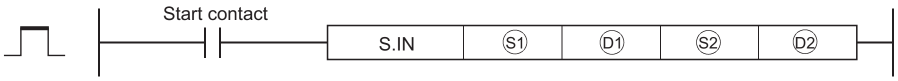
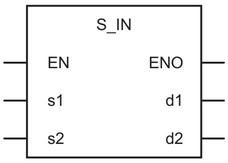
159

9) Processing details of the instruction

10) Error conditions and error codes
For errors other than described here, refer to the QCPU User's Manual (Hardware Design, Maintenance and Inspection).

CHAPTER 8 I/O CONTROL INSTRUCTIONS

8.1 S.IN

<table>
<tr><th colspan="2">Ladder diagram</th></tr>
<tr><td colspan="2">S.IN 
</td></tr>
<tr><th>Structured ladder/FBD</th><th>Structured text language</th></tr>
<tr><td>
</td><td>ENO:=S_IN(EN,s1,s2,d1,d2);</td></tr>
</table>

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Real data type
	S2	: Operation constant start device	: Array of real data type (0..7)
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Loop tag memory start device	: Array of any 16-bit data (0..127)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

Function

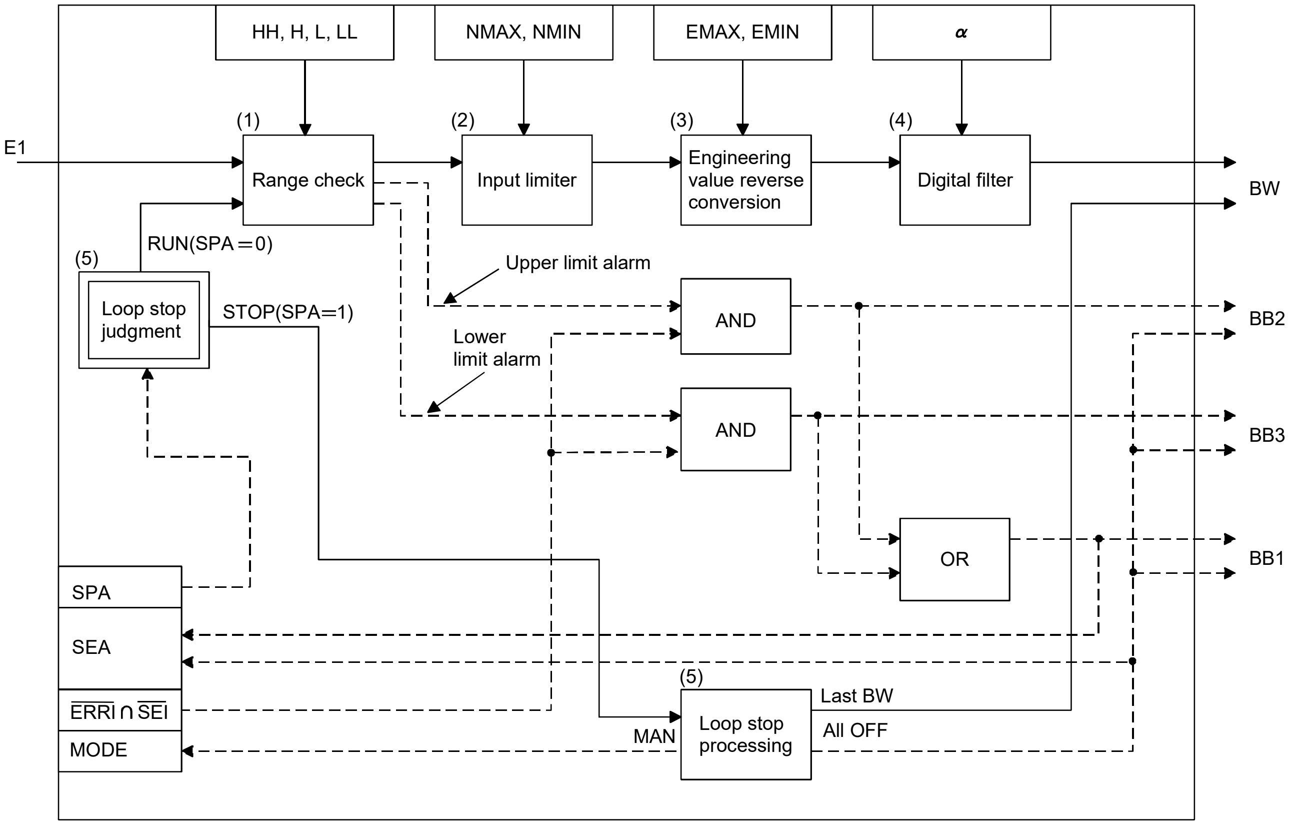
Converts the input value (E1) of the device specified in S1 into an engineering value, and stores the result into the device specified in D1.

Also performs the range check, input limiter processing and digital filter processing of the input value (E1) at this time.

Block diagram

The processing block diagram of the S.IN instruction is shown below.

(The numerals (1) to (5) in the diagram indicate the order of the processing.)

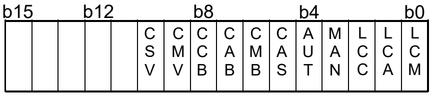
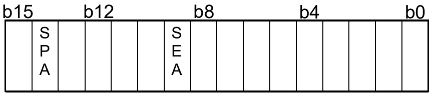
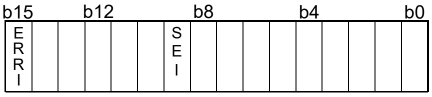


Set Data

(1) Data specified in S.IN instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value	-999999 to 999999	—	Real number	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	%	Real number	—	S
	+2	BB	—					
		BB1	Alarm	b15 b12 b8 b4 b0 BB3 BB2 BB1 (0: Without alarm) (1: With alarm)	—	BIN 16bit	—	S
		BB2	Input upper limit alarm					
		BB3	Input lower limit alarm					

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Operation constant	Ⓢ2+0 +1	EMAX	Engineering conversion upper limit	-999999 to 999999	%	Real number	100.0	U
	+2 +3	EMIN	Engineering conversion lower limit	-999999 to 999999	%	Real number	0.0	U
	+4 +5	NMAX	Input upper limit	-999999 to 999999	—	Real number	100.0	U
	+6 +7	NMIN	Input lower limit	-999999 to 999999	—	Real number	0.0	U
	+8 +9	HH	Upper limit range error occurrence	-999999 to 999999	—	Real number	110.0	U
	+10 +11	H	Upper limit range error return	-999999 to 999999	—	Real number	100.0	U
	+12 +13	L	Lower limit range error return	-999999 to 999999	—	Real number	0.0	U
	+14 +15	LL	Lower limit range error occurrence	-999999 to 999999	—	Real number	-10.0	U
Loop tag memory [*2]	Ⓓ1+1	MODE	Operation mode	0 to FFFF$_H$  b15 b12 b8 b4 b0 CSV CMV CCB CAB CMB CAS AUT MAN LCC LCA LCM	—	BIN 16bit	8$_H$	S/U
	+3	ALM	Alarm detection	0 to FFFF$_H$  SPA 0: Loop RUN 1: Loop STOP SEA (0: Without alarm) (1: With alarm)	—	BIN 16bit	4000$_H$	S/U
	+4	INH	Alarm detection inhibition	0 to FFFF$_H$  0: Alarm enable 1: Alarm inhibit	—	BIN 16bit	0$_H$	S/U
	+38 +39	α	Filter coefficient	0 to 1	—	Real number	0.2	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[*2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Processing contents

(1) Range check

(a) A range check is performed on the input value (E1).
An alarm is output if the input value (E1) exceeds the upper/lower limit value.

Range check	Condition	Range check result (alarm output)		
		BB2	BB3	BB1, SEA
Upper limit check	E1 ≥ HH	1 [*1]	—	1 [*1]
	E1 ≤ H	0	—	0
	H < E1 < HH	Last value	—	Last value
Lower limit check	E1 ≤ LL	—	1 [*1]	1 [*1]
	E1 ≥ L	—	0	0
	LL < E1 < L	—	Last value	Last value

[*1] When SEI or ERRI in the alarm detection inhibition (INH) is set to 1, SEA, BB1, BB2 and BB3 show 0 since the alarm is prohibited.

(b) Last value hold processing
When a range excess occurs (BB1 = 1) in the range check, whether operation will be continued unchanged or the S.IN instruction will be terminated is judged depending on whether SM1500 is ON or OFF.

1) When SM1500 is OFF (not in the hold mode), "(2) Input limiter processing" is performed if a range excess occurs (BB1 = 1).

2) When SM1500 is ON (in the hold mode), the following processing is performed and the S.IN instruction is terminated if a range excess occurs (BB1 = 1).

- BW retains the last value.
- Error information is set in BB.

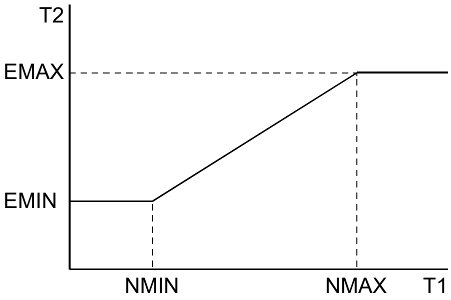
(2) Input limiter processing
Upper/lower limiter setting is made on the input value (E1).

Condition	Result (T1)
E1 ≥ NMAX	NMAX
E1 ≤ NMIN	NMIN
NMIN < E1 < NMAX	E1

(3) Engineering value reverse conversion

The result (T1) of the input limiter is converted reversely from the engineering value according to the following expression.

$$T2 = (EMAX - EMIN) \times \frac{T1 - NMIN}{NMAX - NMIN} + EMIN$$



(4) Digital filter

The input value (E1) is digitally filtered according to the following expression.

The digital filter is used to reduce noise.

BW=T2 + α × (Previous BW value - T2)

(5) Loop stop processing

(a) Setting 1 in SPA of the alarm detection (ALM) selects a loop stop.

A loop stop performs the following processing and terminates the S.IN instruction.

1) BW retains the last value.

2) SEA of the alarm detection (ALM) are turned to 0.

3) The operation mode (MODE) is changed to MAN.

4) BB1 to BB3 of BB are turned to 0.

(b) Setting 0 in SPA of the alarm detection (ALM) selects a loop run.

A loop run performs "(1) Range check".

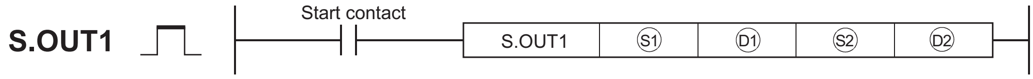
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

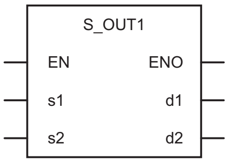
Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓓ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	○	○	—
	When HH < H, L < LL, or NMAX < NMIN	○	○	○
4140	When the values of Ⓢ1, Ⓓ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

8.2 S.OUT1

Ladder diagram



Structured ladder/FBD



Structured text language

```
ENO:=S_OUT1(EN,s1,s2,d1,d2);
```

Input argument	EN	: Execution condition	: Bit
	Ⓢ1	: Input data start device	: Real data type
	Ⓢ2	: Operation constant start device	: Array of real data type (0..1)
Output argument	ENO	: Execution result	: Bit
	Ⓓ1	: Block memory start device	: Array of any 16-bit data (0..2)
	Ⓓ2	: Loop tag memory start device	: Array of any 16-bit data (0..127)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
Ⓢ1	—	○		—					
Ⓓ1	—	○		—					
Ⓢ2	—	○		—					
Ⓓ2	—	○		—					

Function

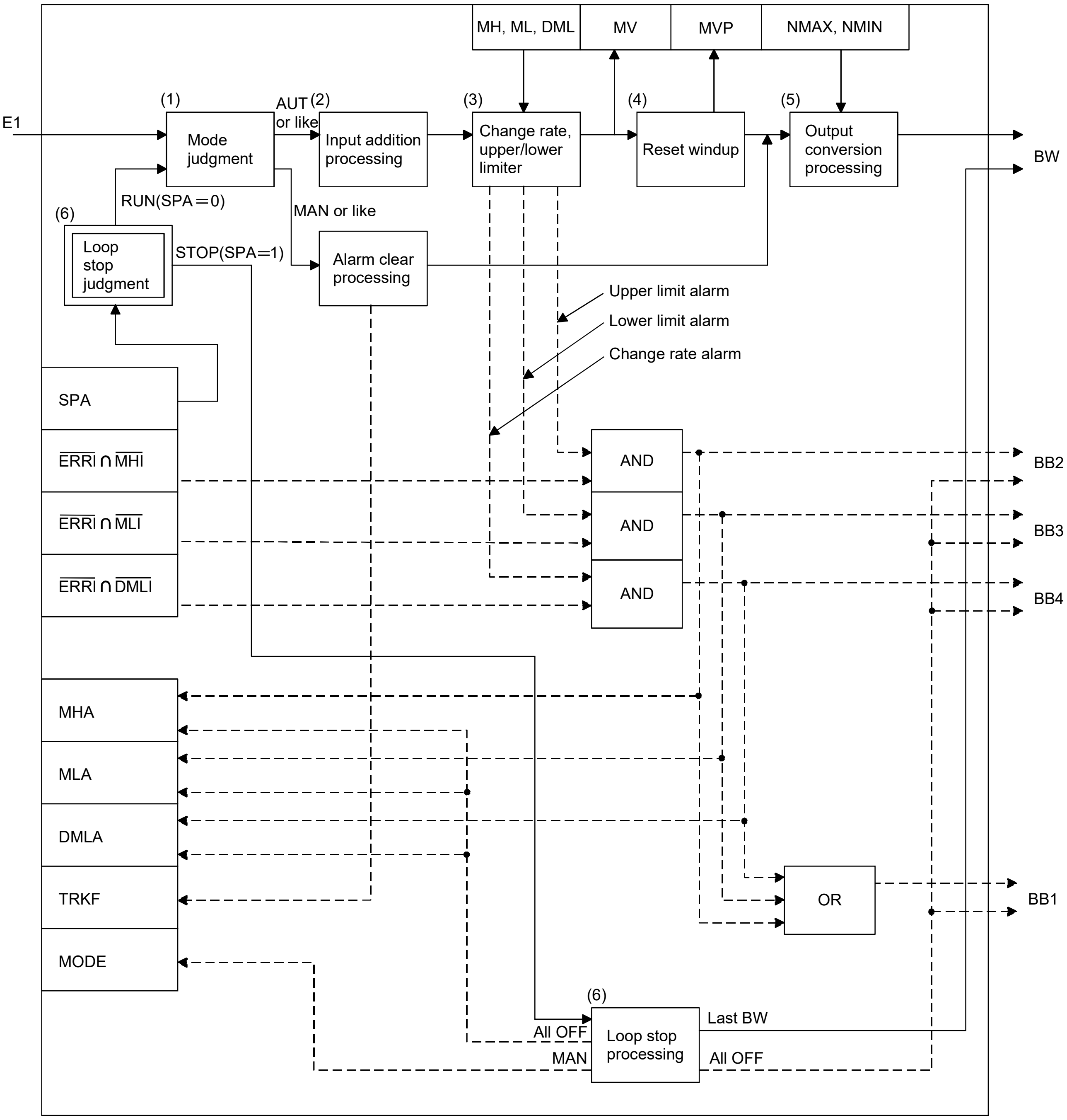
Calculates the manipulated value (MV) by performing input addition processing from the input value (E1 = ΔMV) of the device specified in Ⓢ1, and stores the result into the device specified in Ⓓ1.

Also performs the change rate, upper/lower limiter, reset windup and output conversion processings of the calculated manipulated value (MV) at this time.

Block diagram

The processing block diagram of the S.OUT1 instruction is shown below.

(The numerals (1) to (6) in the diagram indicate the order of the processing.)



Set Data

(1) Data specified in S.OUT1 instruction

Specified position		Symbol	Name	Recommended range *1	Unit	Data format	Standard value	Set by
Input data	S1+0 +1	E1	Input value (ΔMV)	-999999 to 999999	%	Real number	—	U
Block memory	D1+0 +1	BW	Output value	(-999999 to 999999)	—	Real number	—	S
	+2	BB	—					
		BB1	Alarm	b15 b12 b8 b4 b0 b15–b4: (empty), b3: BB4, b2: BB3, b1: BB2, b0: BB1 (0: Without alarm) (1: With alarm)	—	BIN 16bit	—	S
		BB2	Output upper limit alarm					
		BB3	Output lower limit alarm					
		BB4	Output change rate alarm					
Operation constant	S2+0 +1	NMAX	Output conversion upper limit	-999999 to 999999	—	Real number	100.0	U
	+2 +3	NMIN	Output conversion lower limit	-999999 to 999999	—	Real number	0.0	U
Loop tag memory *2	D2+1	MODE	Operation mode	0 to FFFF$_H$ b15 b12 b8 b4 b0 b15–b11: (empty), b10: CSV, b9: CMV, b8: CCB, b7: CAB, b6: CMB, b5: CAS, b4: AUT, b3: MAN, b2: LCC, b1: LCA, b0: LCM	—	BIN 16bit	8$_H$	S/U
	+3	ALM	Alarm detection	0 to FFFF$_H$ b15 b12 b8 b4 b0 b15: (empty), b14: SPA, b13–b12: (empty), b11: DMLA, b10: (empty), b9: SEA, b8–b2: (empty), b1: MHA, b0: MLA SPA 0: Loop RUN 1: Loop STOP DMLA, SEA, MHA, MLA (0: Without alarm) (1: With alarm)	—	BIN 16bit	4000$_H$	S/U
	+4	INH	Alarm detection inhibition	0 to FFFF$_H$ b15 b12 b8 b4 b0 b15: ERRI, b14: (empty), b13: TRKF, b12: (empty), b11: DMLI, b10–b2: (empty), b1: MHI, b0: MLI TRKF (0: Without tracking) (1: With tracking) ERRI, DMLI, MHI, MLI 0: Alarm enable 1: Alarm inhibit	—	BIN 16bit	0$_H$	S/U
	+12 +13	MV	Manipulated value	-10 to 110	%	Real number	0.0	S/U

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

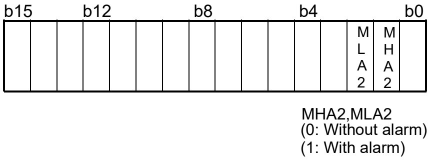
*2 The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Specified position		Symbol	Name	Recommended range [1]	Unit	Data format	Standard value	Set by
Loop tag memory [2]	Ⓓ2+18 +19	MH	Output upper limit value	-10 to 110	%	Real number	100.0	U
	+20 +21	ML	Output lower limit value	-10 to 110	%	Real number	0.0	U
	+48 +49	DML	Output change rate limit value	0 to 100	%	Real number	100.0	U
	+54 +55	I	Integral constant	0 to 999999	s	Real number	10.0	U
	+62 +63	MVP	MV inside operation value	(-999999 to 999999)	%	Real number	0.0	S
Loop tag past value memory [2] [3]	Ⓓ2+116	—	—	Used by the system as a work area.	—	—	—	S

[1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

[3] The application of the loop tag past value memory are indicated below.

Specified position	Description
Ⓓ2 +116	Alarm detection 2 (ALM2)  MHA2,MLA2 (0: Without alarm) (1: With alarm)

When control is to be started from the initial status, the data must be cleared with the sequence program.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) Mode judgment

Either of the following processings is performed depending on the operation mode (MODE).

(a) When the operation mode (MODE) is any of MAN, CMB, CMV and LCM (alarm clear processing)

1) MHA, MLA and DMLA of the alarm detection (ALM) are turned to 0.

2) MHA2 and MLA2 of the alarm detection 2 (ALM2) are turned to 0.

3) BB1 to BB4 of BB are turned to 0.

4) The tracking flag (TRKF) of the alarm detection inhibition (INH) is turned to 1.

5) "(5) Output conversion processing" is performed and the instruction is terminated.

(b) When the operation mode (MODE) is any of AUT, CAB, CAS, CCB, CSV, LCA and LCC, "(2) Input addition processing" is performed.

However, when SEA of the alarm detection (ALM) is 1 and SM1501 is ON (with hold), BB1 to BB4 are turned to 0 and the S.OUT1 instruction is terminated.

(2) Input addition processing

The temporary MV (T) is calculated on the basis of the input value (E1 = ΔMV).

(a) When the tracking flag (TRKF) of the alarm detection inhibition (INH) is 1, the following processing is performed.

1) The manipulated value (MV) is stored into the MV internal operation value (MVP).

2) The input value (E1) is changed to 0. (ΔMV = 0)

3) The tracking flag (TRKF) of the alarm detection inhibition (INH) is turned to 0.

4) The temporary MV (T) is calculated with the following expression.

T = E1 + MVP

MVP = T

(b) When the tracking flag (TRKF) of the alarm detection inhibition (INH) is 0, the temporary MV (T) is calculated with the following expression.

T = E1 + MVP

MVP = T

(3) Change rate, upper/lower limiter

The change rate and upper/lower limits are checked for the input value (E1 = ΔMV), and the data after the limiter processing and an alarm are output.

(a) The change rate limiter performs the following operation and outputs the result of the operation to BB4 and DMLA.

Condition	BB4, DMLA	Result (T1)
$\lvert T - MV\rvert \leq DML$	0	T
(T - MV) > DML	1 [*1]	MV + DML
(T - MV) < - DML	1 [*1]	MV - DML

*1 When DMLI or ERRI in the alarm detection inhibition (INH) is set to 1, DMLA and BB4 show 0 since the alarm is prohibited.

(b) The upper/lower limiter performs the following operation and outputs the result of the operation to BB2, BB3, MHA, MLA, MHA2, and MLA2.

Condition	BB3, MLA, MLA2	BB2, MHA, MHA2	MV
T1 > MH	0	1 [*2]	MH
T1 < ML	1 [*3]	0	ML
ML ≤ T1 ≤ MH	0	0	T1

*2 When MHI or ERRI in the alarm detection inhibition (INH) is set to 1, MHA and BB2 show 0 since the alarm is prohibited. However, even if MHI and/or ERRI in the alarm detection inhibition (INH) is set to 1, MHA2 holds 1.

*3 When MLI or ERRI in the alarm detection inhibition (INH) is set to 1, MLA and BB3 show 0 since the alarm is prohibited. However, even if MLI and/or ERRI in the alarm detection inhibition (INH) is set to 1, MLA2 holds 1.

(4) Reset windup

If the manipulated value (MV) exceeds the upper/lower limit value, the following operation is performed to return it to the upper/lower limit value and enable immediate response when the deviation is inverted.

However, when the integral constant (T1) is 0, the reset windup processing is not performed.

Condition	Operation expression
When T1 > MH, $\frac{\Delta T}{T_I} \leqq 1$	$MVP = \frac{\Delta T}{T_I}(MH - T) + T$
When T1 < ML, $\frac{\Delta T}{T_I} \leqq 1$	$MVP = \frac{\Delta T}{T_I}(ML - T) + T$

(5) Output conversion

In the output conversion, the output value is calculated from the following formula.

$$BW = \frac{NMAX - NMIN}{100} \times MV + NMIN$$

(6) Loop stop processing

(a) Setting 1 in SPA of the alarm detection (ALM) selects a loop stop.

A loop stop performs the following processing and terminates the S.OUT1 instruction.

1) BW retains the last value.

2) DMLA, MHA and MLA of the alarm detection (ALM) are turned to 0.

3) MHA2 and MLA2 of the alarm detection 2 (ALM2) are turned to 0.

4) The operation mode (MODE) is changed to MAN.

5) BB1 to BB4 of BB are turned to 0.

(b) Setting 0 in SPA of the alarm detection (ALM) selects a loop run.

A loop run performs "(1) Mode judgment".

(7) Hold processing

Used to specify whether the output value will be held or not by the S.OUT1 instruction at sensor error occurrence (detected by the S.IN instruction).

A hold processing is performed when the value is determined as RUN at "Loop stop judgement".

Use SM1501 to select whether the manipulated value (MV) will be held or not at sensor alarm occurrence.

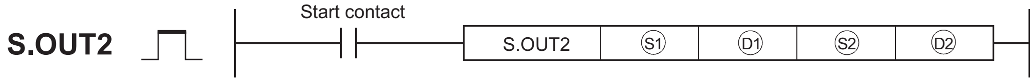
- SM1501 = OFF: Manipulated value (MV) will not be held.
- SM1501 = ON: Manipulated value (MV) will be held.

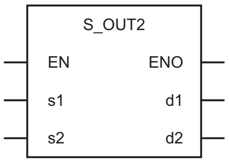
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of S1, S2, D2 are either a non-numeric or non-normalized number	○	○	—
4140	When the values of S1, S2, D2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

8.3 S.OUT2

Ladder diagram


Structured ladder/FBD	Structured text language
	ENO:=S_OUT2(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Real data type
	S2	: Operation constant start device	: Array of real data type (0..1)
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Loop tag memory start device	: Array of any 16-bit data (0..127)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

Function

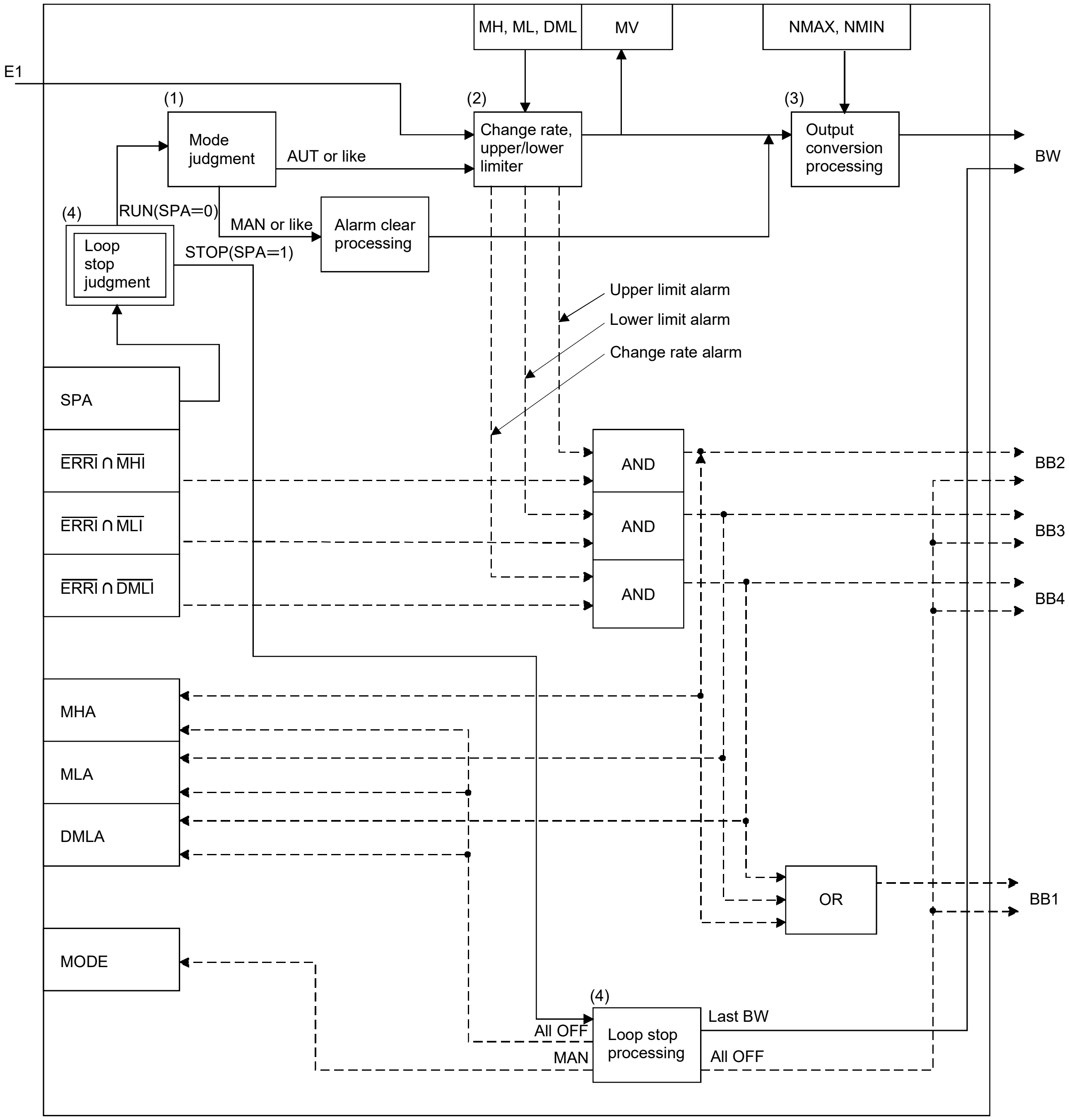
Converts the input value (E1 = MV) of the device specified in S1 into an output, and stores the result into the device specified in D1.

Also performs the change rate, upper/lower limiter processing and output conversion processing of the input value at this time.

Block diagram

The processing block diagram of the S.OUT2 instruction is shown below.

(The numerals (1) to (4) in the diagram indicate the order of the processing.)



Set Data

(1) Data specified in S.OUT2 instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value(MV)	-999999 to 999999	%	Real number	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	—	Real number	—	S
	+2	BB						
		BB1	Alarm	b15 b12 b8 b4 b0 BB4 BB3 BB2 BB1 (0: Without alarm) (1: With alarm)	—	BIN 16bit	—	S
		BB2	Output upper limit alarm					
		BB3	Output lower limit alarm					
		BB4	Output change rate alarm					
Operation constant	Ⓢ2+0 +1	NMAX	Output conversion upper limit	-999999 to 999999	—	Real number	100.0	U
	+2 +3	NMIN	Output conversion lower limit	-999999 to 999999	—	Real number	0.0	U
Loop tag memory [*2]	Ⓓ2+1	MODE	Operation mode	0 to FFFF$_H$ b15 b12 b8 b4 b0 CSV CMV CCB CAB CMB CAS AUT MAN LCC LCA LCM	—	BIN 16bit	8$_H$	S/U
	+3	ALM	Alarm detection	0 to FFFF$_H$ b15 b12 b8 b4 b0 SPA DMLA SEA MHA MLA SPA 0: Loop RUN 1: Loop STOP DMLA, SEA, MHA, MLA (0: Without alarm) (1: With alarm)	—	BIN 16bit	4000$_H$	S/U
	+4	INH	Alarm detection inhibition	0 to FFFF$_H$ b15 b12 b8 b4 b0 ERRI DMLI MHI MLI 0: Alarm enable 1: Alarm inhibit	—	BIN 16bit	0$_H$	S/U
	+12 +13	MV	Manipulated value	-10 to 110	%	Real number	0.0	S/U
	+18 +19	MH	Output upper limit value	-10 to 110	%	Real number	100.0	U
	+20 +21	ML	Output lower limit value	-10 to 110	%	Real number	0.0	U
	+48 +49	DML	Output change rate limit value	0 to 100	%	Real number	100.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[*2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Processing contents

(1) Mode judgment

Either of the following processings is performed depending on the operation mode (MODE).

(a) When the operation mode (MODE) is any of MAN, CMB, CMV and LCM (alarm clear processing)

1) MHA, MLA and DMLA of the alarm detection (ALM) are turned to 0.

2) BB1 to BB4 of BB are turned to 0.

3) "(3) Output conversion processing" is performed and the S.OUT2 instruction is terminated.

(b) When the operation mode (MODE) is any of AUT, CAB, CAS, CCB, CSV, LCA and LCC, "(2) Change rate, upper/lower limiter" is performed.

However, when SEA of the alarm detection (ALM) is 1 and SM1501 is ON (with hold), BB1 to BB4 are turned to 0 and the S.OUT2 instruction is terminated.

(2) Change rate, upper/lower limiter

The change rate and upper/lower limits are checked for the input value (E1), and the data after the limiter processing and an alarm are output.

(a) The change rate limiter performs the following operation and outputs the result of the operation to BB4 and DMLA.

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 When DMLI or ERRI in the alarm detection inhibition (INH) is set to 1, DMLA and BB4 show 0 since the alarm is prohibited.

(b) The upper/lower limiter performs the following operation and outputs the result of the operation to BB2, BB3, MHA and MLA.

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 When MHI or ERRI in the alarm detection inhibition (INH) is set to 1, MHA and BB2 show 0 since the alarm is prohibited.

*3 When MLI or ERRI in the alarm detection inhibition (INH) is set to 1, MLA and BB3 show 0 since the alarm is prohibited.

(3) Output conversion

In the output conversion, the output value is calculated from the following formula.

$$BW = \frac{NMAX - NMIN}{100} \times MV + NMIN$$

(4) Loop stop processing

(a) Setting 1 in SPA of the alarm detection (ALM) selects a loop stop.
A loop stop performs the following processing and terminates the S.OUT2 instruction.

1) BW retains the last value.
2) DMLA, MHA and MLA of the alarm detection (ALM) are turned to 0.
3) The operation mode (MODE) is changed to MAN.
4) BB1 to BB4 of BB are turned to 0.

(b) Setting 0 in SPA of the alarm detection (ALM) selects a loop run.
A loop run performs "(1) Mode judgment".

(5) Hold processing

Used to specify whether the output value will be held or not by the S.OUT2 instruction at sensor error occurrence (detected by the S.IN instruction).

A hold processing is performed when the value is determined as RUN at "Loop stop judgement".

Use SM1501 to select whether the manipulated value (MV) will be held or not at sensor alarm occurrence.

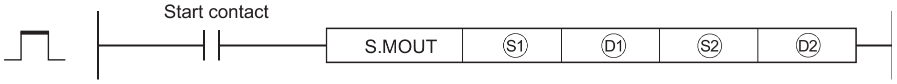
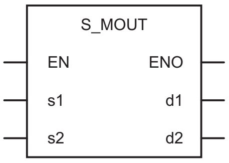
- SM1501 = OFF: Manipulated value (MV) will not be held.
- SM1501 = ON: Manipulated value (MV) will be held.

Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	○	○	—
4140	When the values of Ⓢ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

8.4 S.MOUT

Ladder diagram	
S.MOUT 	
Structured ladder/FBD	**Structured text language**
	ENO:=S_MOUT(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	Ⓢ1	: Dummy device[*1]	: Real data type
	Ⓢ2	: Operation constant start device	: Array of real data type (0..1)
Output argument	ENO	: Execution result	: Bit
	Ⓓ1	: Operation constant start device	: Real data type
	Ⓓ2	: Loop tag memory start device	: Array of any 16-bit data (0..127)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
Ⓢ1	—	○		—					
Ⓓ1	—	○		—					
Ⓢ2	—	○		—					
Ⓓ2	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

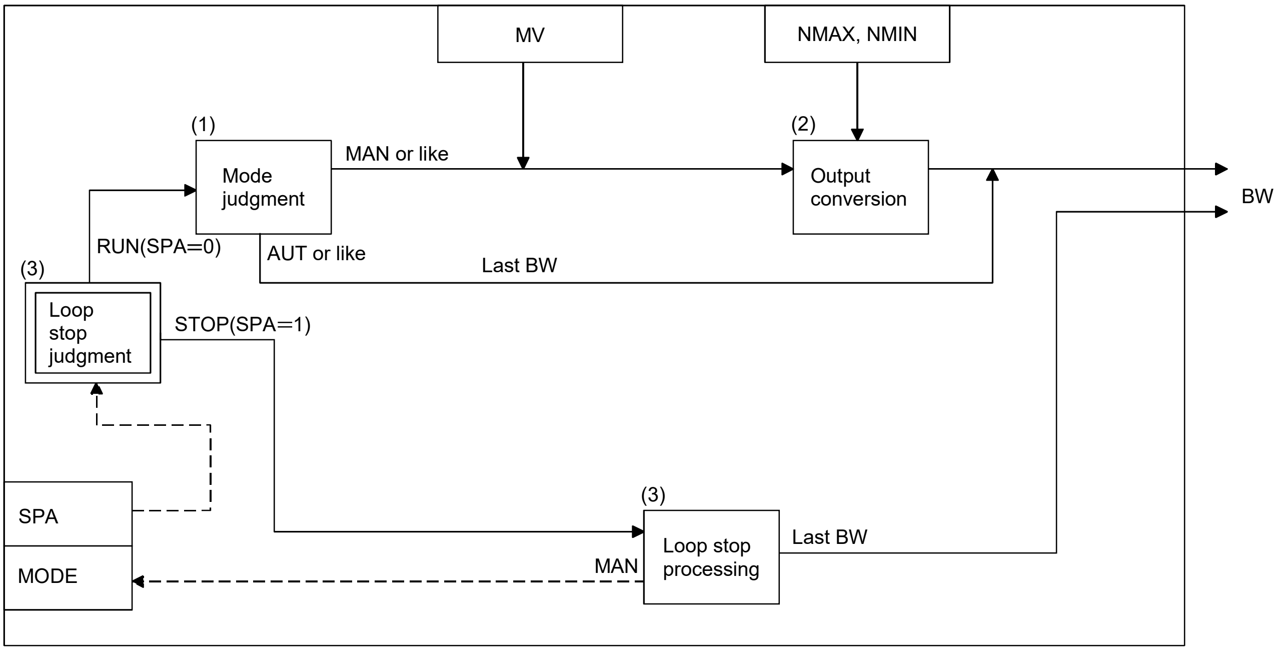
Function

Converts the manipulated value (MV) specified in Ⓓ2 into an output, and stores the result into the device specified in Ⓓ1.

Block diagram

The processing block diagram of the S.MOUT instruction is shown below.

(The numerals (1) to (3) in the diagram indicate the order of the processing.)



Set Data

(1) Data specified in S.MOUT instruction

<table>
<tr><th colspan="2">Specified position</th><th>Symbol</th><th>Name</th><th>Recommended range [*1]</th><th>Unit</th><th>Data format</th><th>Standard value</th><th>Set by</th></tr>
<tr><td>Block memory</td><td>Ⓓ1+0
+1</td><td>BW</td><td>Output value</td><td>(-999999 to 999999)</td><td>—</td><td>Real number</td><td>—</td><td>S</td></tr>
<tr><td rowspan="2">Operation constant</td><td>Ⓢ2+0
+1</td><td>NMAX</td><td>Output conversion upper limit</td><td>-999999 to 999999</td><td>—</td><td>Real number</td><td>100.0</td><td>U</td></tr>
<tr><td>+2
+3</td><td>NMIN</td><td>Output conversion lower limit</td><td>-999999 to 999999</td><td>—</td><td>Real number</td><td>0.0</td><td>U</td></tr>
<tr><td rowspan="3">Loop tag memory [*2]</td><td>Ⓓ2+1</td><td>MODE</td><td>Operation mode</td><td>0 to FFFF$_H$
<table>
<tr><td>b15</td><td></td><td></td><td>b12</td><td></td><td></td><td></td><td>b8</td><td></td><td></td><td></td><td>b4</td><td></td><td></td><td></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>LCC</td><td>LCA</td><td>LCM</td></tr>
</table></td><td>—</td><td>BIN 16bit</td><td>8$_H$</td><td>S/U</td></tr>
<tr><td>+3</td><td>ALM</td><td>Alarm detection</td><td>0 to FFFF$_H$
<table>
<tr><td>b15</td><td></td><td></td><td>b12</td><td></td><td></td><td></td><td>b8</td><td></td><td></td><td></td><td>b4</td><td></td><td></td><td></td><td>b0</td></tr>
<tr><td></td><td>SPA</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
SPA
0: Loop RUN
1: Loop STOP</td><td>—</td><td>BIN 16bit</td><td>4000$_H$</td><td>S/U</td></tr>
<tr><td>+12
+13</td><td>MV</td><td>Manipulated value</td><td>-10 to 110</td><td>%</td><td>Real number</td><td>0.0</td><td>U</td></tr>
</table>

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[*2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Processing contents

(1) Mode judgment

Either of the following processings is performed depending on the operation mode (MODE).

(a) When the operation mode (MODE) is any of MAN, CMB, CMV and LCM

1) The manipulated value (MV) is used as the output value (BW).

2) "(2) Output conversion processing" is performed.

(b) When the operation mode (MODE) is any of AUT, CAB, CAS, CCB, CSV, LCA and LCC, BW retains the last value.

(2) Output conversion

In the output conversion, the output value is calculated from the following formula.

$$BW = \frac{NMAX - NMIN}{100} \times MV + NMIN$$

8

8.4 S.MOUT

(3) Loop stop processing

(a) Setting 1 in SPA of the alarm detection (ALM) selects a loop stop.
A loop stop performs the following processing and terminates the S.MOUT instruction.

1) BW retains the last value.

2) The operation mode (MODE) is changed to MAN.

(b) Setting 0 in SPA of the alarm detection (ALM) selects a loop run.
A loop run performs "(1) Mode judgment".

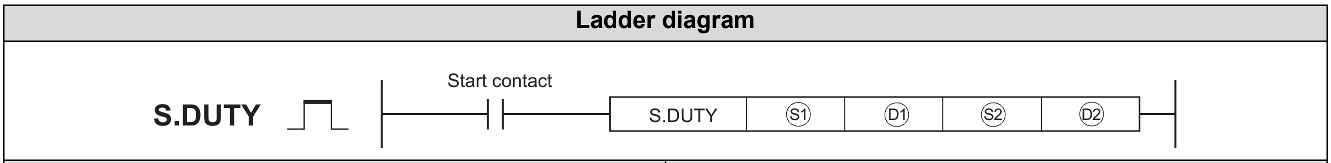
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

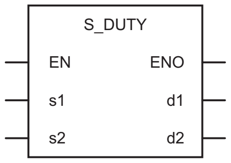
Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of S2, D2 are either a non-numeric or non-normalized number	○	○	—
4140	When the values of S2, D2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

8.5 S.DUTY

Ladder diagram



Structured ladder/FBD



Structured text language

ENO:=S_DUTY(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	Ⓢ1	: Input data start device	: Real data type
	Ⓢ2	: Dummy device*1	: Any 16-bit data
Output argument	ENO	: Execution result	: Bit
	Ⓓ1	: Operation constant start device	: Array of any 16-bit data (0..1)
	Ⓓ2	: Loop tag memory start device	: Array of any 16-bit data (0..127)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
Ⓢ1	—	○		—					
Ⓓ1	—	○		—					
Ⓢ2	—	○		—					
Ⓓ2	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

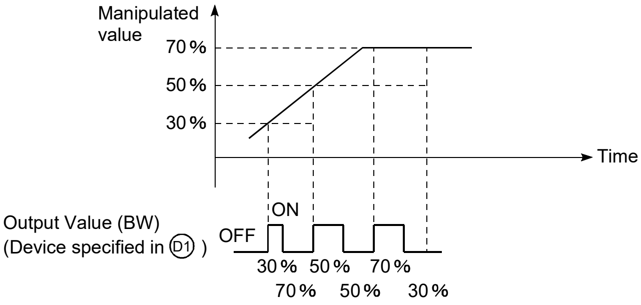
Function

Calculates the manipulated value (MV) by performing input addition processing from the input value (E1 = ΔMV) of the device specified in Ⓢ1. Turns ON/OFF the device specified in Ⓓ1 in proportion to the manipulated value (MV).

The ON/OFF time is a value on the assumption that the time specified as the control output cycle (CTDUTY) is 100%.

The ON/OFF time is changed in each execution cycle.

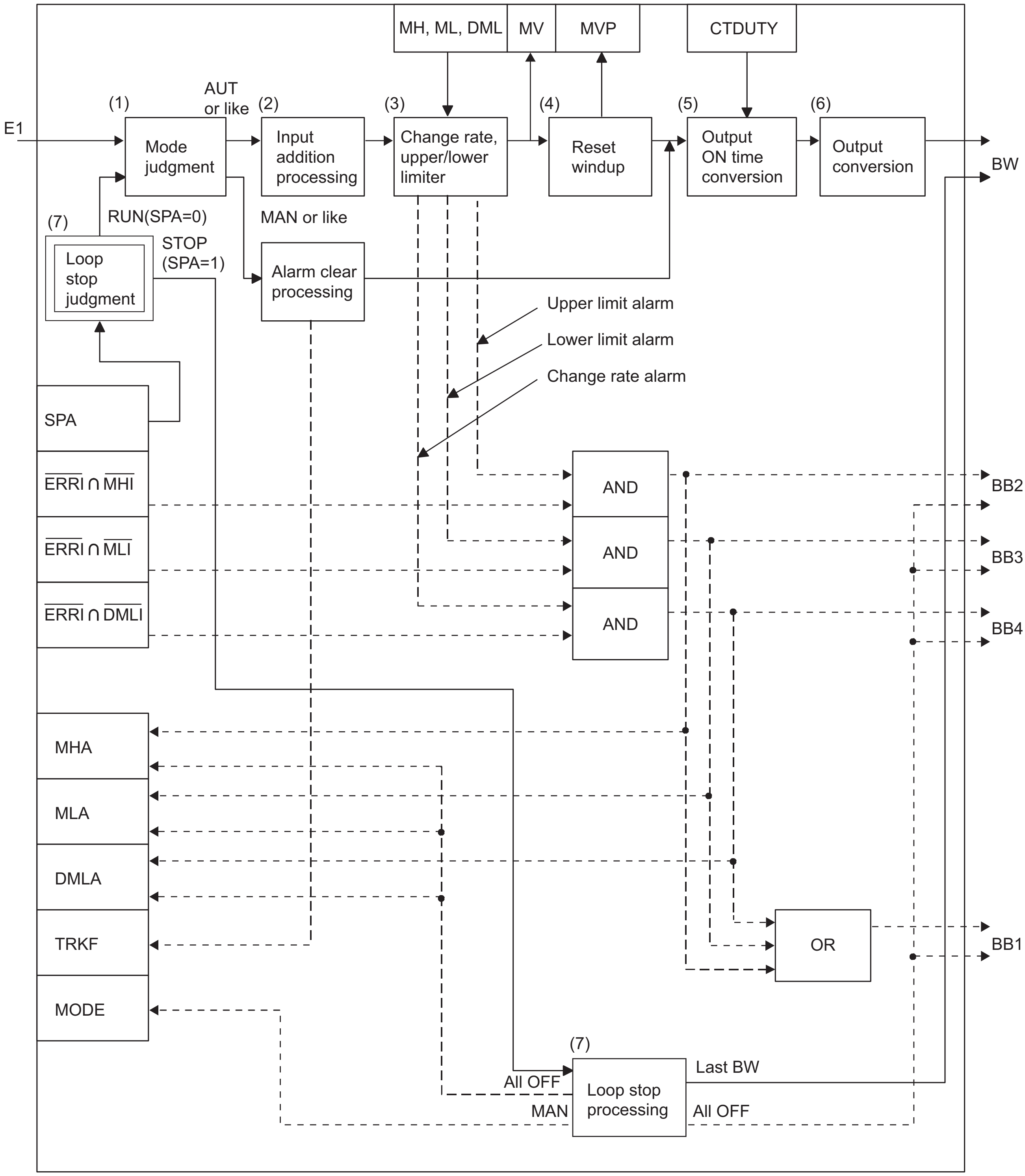
Also performs the change rate, upper/lower limiter and reset windup of the calculated manipulated value (MV) at this time.



Block diagram

The processing block diagram of the S.DUTY instruction is shown below.

(The numerals (1) to (7) in the diagram indicate the order of the processing.)



Set Data

(1) Data specified in S.DUTY instruction

<table>
<tr><th colspan="2">Specified position</th><th>Symbol</th><th>Name</th><th>Recommended range [1]</th><th>Unit</th><th>Data format</th><th>Standard value</th><th>Set by</th></tr>
<tr><td>Input data</td><td>S1+0
+1</td><td>E1</td><td>Input value (ΔMV)</td><td>-999999 to 999999</td><td>%</td><td>Real number</td><td>—</td><td>U</td></tr>
<tr><td rowspan="7">Block memory</td><td rowspan="2">D1+0</td><td>BW</td><td colspan="6">—</td></tr>
<tr><td>BW1</td><td>Output bit</td><td><table><tr><td>b15</td><td></td><td></td><td>b12</td><td></td><td></td><td></td><td>b8</td><td></td><td></td><td></td><td>b4</td><td></td><td></td><td></td><td>b0</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>BW1</td></tr></table>(0: OFF)
(1: ON)</td><td>—</td><td>BIN 16Bit</td><td>—</td><td>S</td></tr>
<tr><td rowspan="5">+1</td><td>BB</td><td colspan="6"></td></tr>
<tr><td>BB1</td><td>Alarm</td><td rowspan="4"><table><tr><td>b15</td><td></td><td></td><td>b12</td><td></td><td></td><td></td><td>b8</td><td></td><td></td><td></td><td>b4</td><td></td><td></td><td></td><td>b0</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>BB4</td><td>BB3</td><td>BB2</td><td>BB1</td></tr></table>(0: Without alarm)
(1: With alarm)</td><td rowspan="4">—</td><td rowspan="4">BIN 16Bit</td><td rowspan="4">—</td><td rowspan="4">S</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 change rate alarm</td></tr>
<tr><td rowspan="4">Loop tag memory [2]</td><td>D2+1</td><td>MODE</td><td>Operation mode</td><td>0 to $FFFF_H$<table><tr><td>b15</td><td></td><td></td><td>b12</td><td></td><td></td><td></td><td>b8</td><td></td><td></td><td></td><td>b4</td><td></td><td></td><td></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>LCC</td><td>LCA</td><td>LCM</td></tr></table></td><td>—</td><td>BIN 16Bit</td><td>8_H</td><td>S/U</td></tr>
<tr><td>+3</td><td>ALM</td><td>Alarm detection</td><td>0 to $FFFF_H$<table><tr><td>b15</td><td></td><td></td><td>b12</td><td></td><td></td><td></td><td>b8</td><td></td><td></td><td></td><td>b4</td><td></td><td></td><td></td><td>b0</td></tr><tr><td></td><td>SPA</td><td></td><td></td><td>DMLA</td><td></td><td>SEA</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>MHA</td><td>MLA</td></tr></table>SPA
0: Loop RUN
1: Loop STOP
DLMA, SEA, MHA, MLA
(0: Without alarm)
(1: With alarm)</td><td>—</td><td>BIN 16Bit</td><td>4000_H</td><td>S/U</td></tr>
<tr><td>+4</td><td>INH</td><td>Alarm detection inhibition</td><td>0 to $FFFF_H$<table><tr><td>b15</td><td></td><td></td><td>b12</td><td></td><td></td><td></td><td>b8</td><td></td><td></td><td></td><td>b4</td><td></td><td></td><td></td><td>b0</td></tr><tr><td>ERRI</td><td></td><td>TRKF</td><td></td><td>DMLI</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>MHI</td><td>MLI</td></tr></table>TRKF
(0: Without tracking)
(1: With tracking)
ERRI, DMLI, MHI, MLI
0: Alarm enable
1: Alarm inhibit</td><td>—</td><td>BIN 16Bit</td><td>0_H</td><td>S/U</td></tr>
<tr><td>+12
+13</td><td>MV</td><td>Manipulated value</td><td>-10 to 110</td><td>%</td><td>Real number</td><td>0.0</td><td>S/U</td></tr>
</table>

[1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

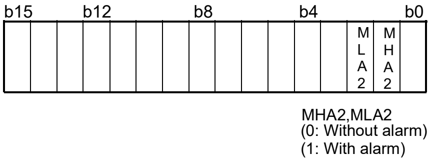
[2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Loop tag memory [*2]	Ⓓ2+18 +19	MH	Output upper limit value	-10 to 110	%	Real number	100.0	U
	+20 +21	ML	Output lower limit value	-10 to 110	%	Real number	0.0	U
	+48 +49	DML	Output change rate limit value	0 to 100	%	Real number	100.0	U
	+54 +55	I	Integral constant	0 to 999999	s	Real number	10.0	U
	+62 +63	MVP	MV inside operation value	(-999999 to 999999)	%	Real number	0.0	S
	+68 +69	CTDUTY	Control output cycle	0 to 999999 Note that $\frac{\text{CTDUTY}}{\Delta T} \leqq 32767$	s	Real number	1.0	U
Loop tag past value memory *2 *3	Ⓓ2+116 ⋮ +121	—	—	Used by the system as a work area.	—	—	—	S

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

*2 The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

*3 The applications of the loop tag past value memory are indicated below.

Specified position	Description
Ⓓ2 +116	Alarm detection 2 (ALM2)  b15 b12 b8 b4 b0 MLA2 MHA2 MHA2,MLA2 (0: Without alarm) (1: With alarm)
+118	Control output cycle counter initial preset flag
+119	Control output cycle counter[*4]
+120	Output counter
+121	Output ON counter

When control is to be started from the initial status, the data must be cleared with the sequence program.

*4 The control output cycle counter rounds off the data to the nearest whole number.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) Mode judgment

Either of the following processings is performed depending on the operation mode (MODE).

(a) When the operation mode (MODE) is any of MAN, CMB, CMV and LCM (alarm clear processing)

1) MHA, MLA and DMLA of the alarm detection (ALM) are turned to 0.

2) MHA2 and MLA2 of the alarm detection 2 (ALM2) are turned to 0.

3) BB1 to BB4 of BB are turned to 0.

4) The tracking flag (TRKF) of the alarm detection inhibition (INH) is turned to 1.

5) "(5) Output ON time conversion processing" is performed.

(b) When the operation mode (MODE) is any of AUT, CAB, CAS, CCB, CSV, LCA and LCC, "(2) Input addition processing" is performed.

However, when SEA of the alarm detection (ALM) is 1 and SM1501 is ON (with hold), BB1 to BB4 are turned to 0 and the S.DUTY instruction is terminated.

(2) Input addition processing

The temporary MV (T) is calculated on the basis of the input value (E1 = ΔMV).

(a) When the tracking flag (TRKF) of the alarm detection inhibition (INH) is 1, the following processing is performed.

1) The manipulated value (MV) is stored into the MV internal operation value (MVP).

2) The input value (E1) is changed to 0. (ΔMV = 0)

3) The tracking flag (TRKF) of the alarm detection inhibition (INH) is turned to 0.

4) The temporary MV (T) is calculated with the following expression.

T = E1 + MVP
MVP = T

(b) When the tracking flag (TRKF) of the alarm detection inhibition (INH) is 0, the temporary MV (T) is calculated with the following expression.

T = E1 + MVP
MVP = T

(3) Change rate, upper/lower limiter

The change rate and upper/lower limits are checked for a difference between the temporary MV (T) and manipulated value (MV), and the data after the limiter processing and an alarm are output.

(a) The change rate limiter performs the following operation and outputs the result of the operation to BB4 and DMLA.

Condition	BB4, DMLA	Result (T1)
|T - MV| ≤ DML	0	T
(T - MV) > DML	1 [*1]	MV + DML
(T - MV) < - DML	1 [*1]	MV - DML

*1 When DMLI or ERRI in the alarm detection inhibition (INH) is set to 1, DMLA and BB4 show 0 since the alarm is prohibited.

(b) The upper/lower limiter performs the following operation and outputs the result of the operation to BB2, BB3, MHA, MLA, MHA2, and MLA2..

Condition	BB3, MLA, MLA2	BB2, MHA, MHA2	MV
T1 > MH	0	1 [*2]	MH
T1 < ML	1 [*3]	0	ML
ML ≤ T1 ≤ MH	0	0	T1

*2 When MHI or ERRI in the alarm detection inhibition (INH) is set to 1, MHA and BB2 show 0 since the alarm is prohibited. However, even if MHI and/or ERRI in the alarm detection inhibition (INH) is set to1, MHA2 holds 1.

*3 When MLI or ERRI in the alarm detection inhibition (INH) is set to 1, MLA and BB3 show 0 since the alarm is prohibited. However, even if MLI and/or ERRI in the alarm detection inhibition (INH) is set to1, MLA2 holds 1.

(4) Reset windup

If the manipulated value (MV) exceeds the upper/lower limit value, the following operation is performed to return it to the upper/lower limit value and enable immediate response when the deviation is inverted.

However, when the integral constant (T1) is 0, the reset windup processing is not performed.

Condition	Operation expression
When T1 > MH, $\frac{\Delta T}{T_I} \leqq 1$	$MVP = \frac{\Delta T}{T_I}(MH - T) + T$
When T1 < ML, $\frac{\Delta T}{T_I} \leqq 1$	$MVP = \frac{\Delta T}{T_I}(ML - T) + T$

(5) Output ON time conversion processing

(a) When the control output cycle (CTDUTY) is reached, the output ON counter is calculated with the following expression. At this time, the output counter is cleared (to 0).

$$\text{Output ON Counter} = \frac{\text{CTDUTY}}{\Delta T} \times MV \times \frac{1}{100}$$

The output ON counter rounds off a fraction to no decimal places.

(b) When the control output cycle (CTDUTY) is not reached, the output counter is incremented by 1 and "(6) Output conversion processing" is performed.

(6) Output conversion processing

In the output conversion processing, the following processing is performed.

Condition	BW
Output counter < output ON counter	1 (ON)
Output counter ≥output ON counter	0 (OFF)

(7) Loop stop processing

(a) Setting 1 in SPA of the alarm detection (ALM) selects a loop stop.

A loop stop performs the following processing and terminates the S.DUTY instruction.

1) BW is output at the last ON/OFF rate.

2) DMLA, MHA and MLA of the alarm detection (ALM) are turned to 0.

3) MHA2 and MLA2 of the alarm detection 2 (ALM2) are turned to 0.

4) The operation mode (MODE) is changed to MAN.

5) BB1 to BB4 of BB are turned to 0.

(b) Setting 0 in SPA of the alarm detection (ALM) selects a loop run.

A loop run performs "(1) Mode judgment".

(8) Hold processing

Used to specify whether the output value will be held or not by the S. DUTY instruction at sensor error occurrence (detected by the S.IN instruction).

A hold processing is performed when the value is determined as RUN at "Loop stop judgement".

Use SM1501 to select whether the manipulated value (MV) will be held or not at sensor alarm occurrence.

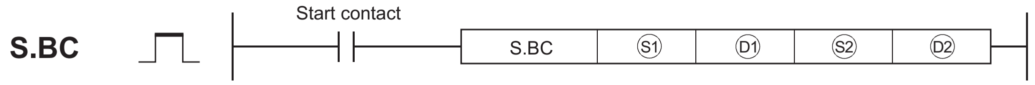
- SM1501 = OFF: Manipulated value (MV) will not be held.
- SM1501 = ON: Manipulated value (MV) will be held.

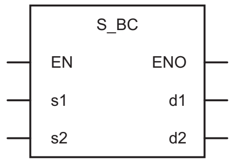
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of S1, D2 are either a non-numeric or non-normalized number	○	○	—
	When CTDUTY < 0 or the execution cycle (SD1500) < 0	○	○	○
4140	When the values of S1, D2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

8.6 S.BC

Ladder diagram


Structured ladder/FBD	Structured text language
	ENO:=S_BC(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Any 32-bit data
	S2	: Dummy device[*1]	: Any 16-bit data
Output argument	ENO	: Execution result	: Bit
	D1	: Operation constant start device	: Array of any 16-bit data (0..1)
	D2	: Loop tag memory start device	: Array of any 16-bit data (0..127)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

Function

Compares the input value (E1) with the set value 1 (SV1)/set value 2 (SV2), and outputs bit data as soon as the input value (E1) reaches the set value 1 (SV1)/set value 2 (SV2).

Also performs the upper limit check processing, change rate check processing and output conversion processing of the input value (E1) at this time.

Set Data

(1) Data specified in S.BC instruction

Specified position		Symbol	Name	Recommended range [1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value	0 to 2147483647	—	BIN 32Bit	—	U
Block memory	Ⓓ1+0	BW		—				
		BW1	Output1	b15 b12 b8 b4 b0 BW2 BW1 (0: OFF) (1: ON)	—	BIN 16Bit	—	S
		BW2	Output2					
	+1	BB		—				
		BB1	Alarm	b15 b12 b8 b4 b0 BB3 BB2 BB1 (0: Without alarm) (1: With alarm)	—	BIN 16Bit	—	S
		BB2	Upper limit alarm					
		BB3	Change rate alarm					
Loop tag memory [2]	Ⓓ2+3	ALM	Alarm detection	0 to $FFFF_H$ b15 b12 b8 b4 b0 PHA DPPA PHA, DPPA (0: Without alarm) (1: With alarm)	—	BIN 16Bit	4000_H	S/U
	+4	INH	Alarm detection inhibition	0 to $FFFF_H$ b15 b12 b8 b4 b0 ERRI PHI DPPI 0: Alarm enable 1: Alarm inhibit	—	BIN 16Bit	0_H	S/U
	+14 +15	SV1	Set value1	0 to 2147483647	—	BIN 32Bit	0	U
	+16 +17	SV2	Set value2	0 to 2147483647	—	BIN 32Bit	0	U
	+26 +27	PH	Upper limit alarm set value	0 to 2147483647	—	BIN 32Bit	0	U
	+42 +43	CTIM	Change rate alarm check time	0 to 999999 Note that $\frac{CTIM}{\Delta T} \leqq 32767$	s	Real number	0.0	U
	+44 +45	DPL	Change rate alarm value	0 to 2147483647	—	BIN 32Bit	0	U

[1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Specified position		Symbol	Name	Recommended range *1	Unit	Data format	Standard value	Set by
Loop tag past value memory *2 *3	Ⓓ2+124	—	—	Used by the system as a work area.	—	—	—	S
	⋮							
	+127							

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

*2 The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

*3 The applications of the loop tag past value memory are indicated below.

Specified position	Description
Ⓓ2 +124	Change rate monitor counter initial preset flag
+125	Change rate monitor counter*4
+126 +127	X_{n-m}

When control is to be started from the initial status, the data must be cleared with the sequence program.

*4 The change rate monitor counter rounds off the data to the nearest whole number.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) Upper limit check processing

In the upper limit check processing, the following operation is performed and the result of the operation is output to BB2 and PHA.

Condition	BB2, PHA
E1 > PH	1 *1
Others	0

*1 When PHI or ERRI in the alarm detection inhibition (INH) is set to 1, PHA and BB2 show 0 since the alarm is prohibited.

(2) Change rate check processing

Performs a change rate alarm check during the change rate alarm check time (CTIM) specified in the loop tag memory. The change rate alarm check compares the change of the input value (E1) with the change rate alarm value (DPL) in each execution cycle (ΔT).

Condition	BB3, DPPA
$(X_n - X_{n-m}) \geq$ DPL	1 [*2]
Others	0

[*2] When DPPI or ERRI of the alarm detection inhibition (INH) is 1, DPPA and BB3 turn to 0 since the alarm is prohibited.

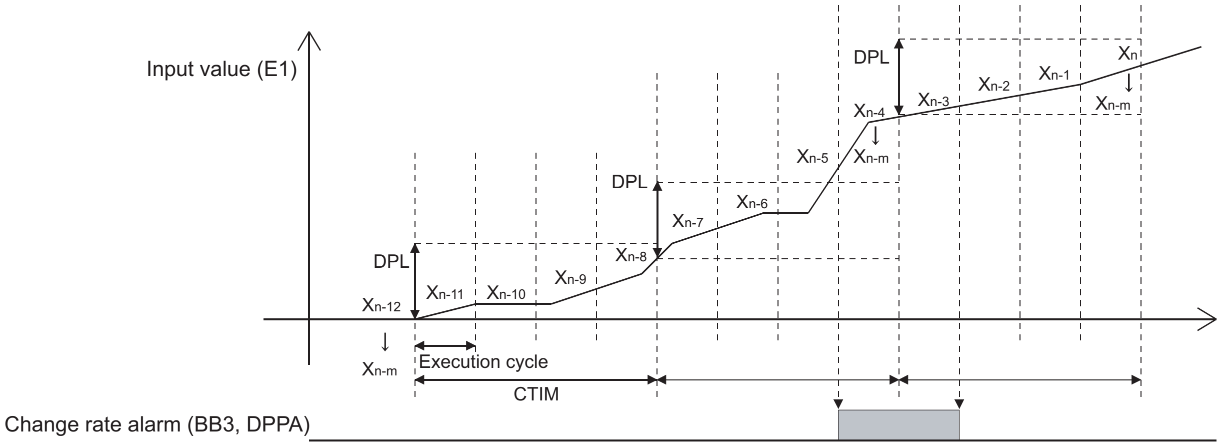
The change rate alarm counter (m) is calculated with the following expression.

$$\text{Change rate alarm counter (m)} = \frac{\text{CTIM}}{\Delta T}$$

Set values of CTIM and ΔT so that the change rate alarm counter (m) ≥ 2.

When the change rate alarm counter (m) = 0, no processing is performed.

Example) When the change rate alarm counter (m) = 4, processing is perform as shown below.



(3) Output conversion processing

In the output conversion processing, the following operation is performed and the result of the operation is stored into BW1 and BW2.

Condition	BW1	BW2
E1 < 0	0	0
0 ≤ E1 < SV1	0	—
E1 ≥ SV1	1	—
0 ≤ E1 < SV2	—	0
E1 ≥ SV2	—	1

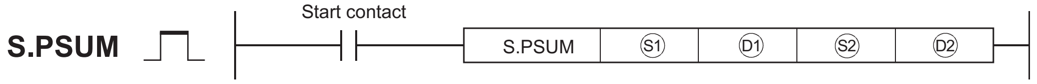
Operation Error

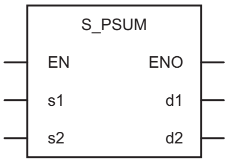
In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of S1, D2 are either a non-numeric or non-normalized number	○	○	—
	When the execution cycle (SD1500) < 0	○	○	○
	When the change rate alarm check time (CTIM) < 0	○	○	○
4140	When the values of S1, D2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

8.7 S.PSUM

Ladder diagram



Structured ladder/FBD	Structured text language
	ENO:=S_PSUM(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	Ⓢ1	: Input data start device	: Array of any 16-bit data (0..2)
	Ⓢ2	: Operation constant start device	: Array of any 16-bit data (0..4)
Output argument	ENO	: Execution result	: Bit
	Ⓓ1	: Block memory start device	: Array of any 32-bit data (0..1)
	Ⓓ2	: Loop tag memory start device	: Array of any 16-bit data (0..127)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
Ⓢ1	—	○		—					
Ⓓ1	—	○		—					
Ⓢ2	—	○		—					
Ⓓ2	—	○		—					

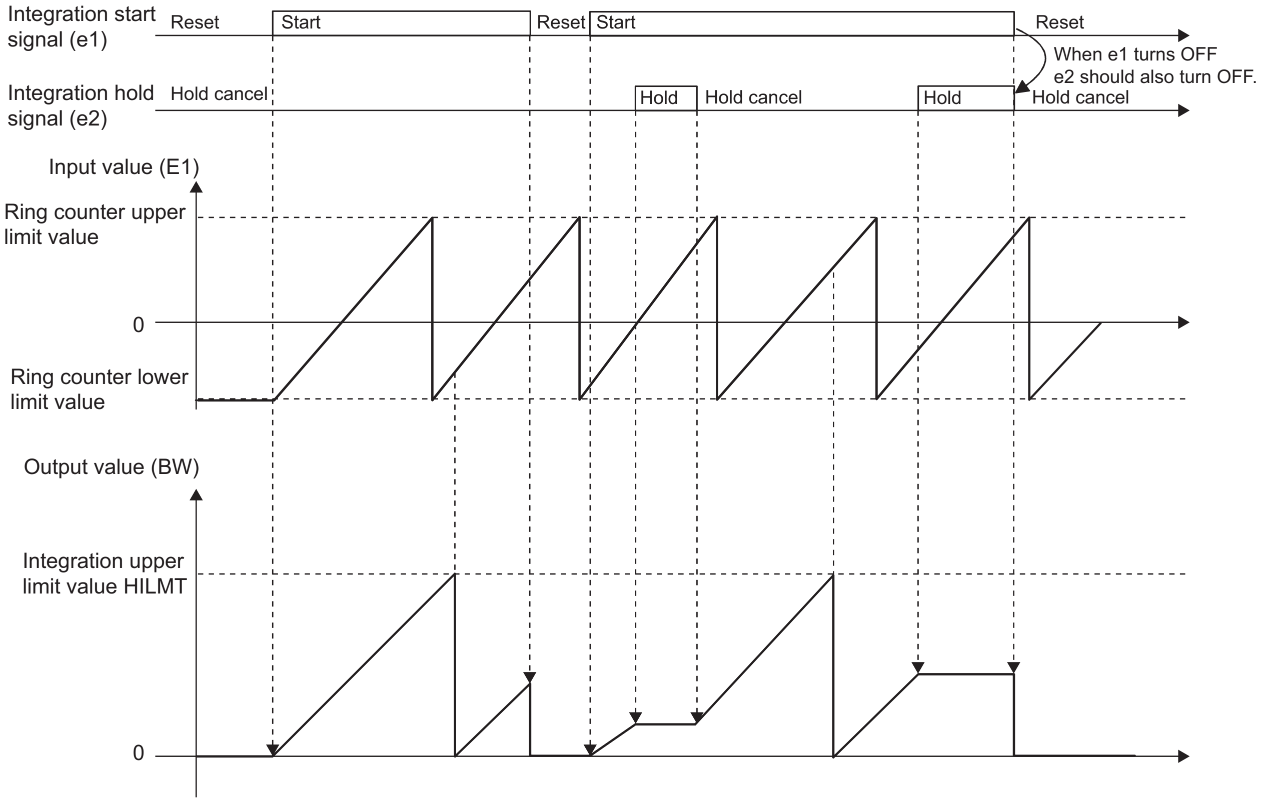
Function

Integrates the input value (E1) of the device specified in Ⓢ1, and stores the result into the device specified in Ⓓ1.

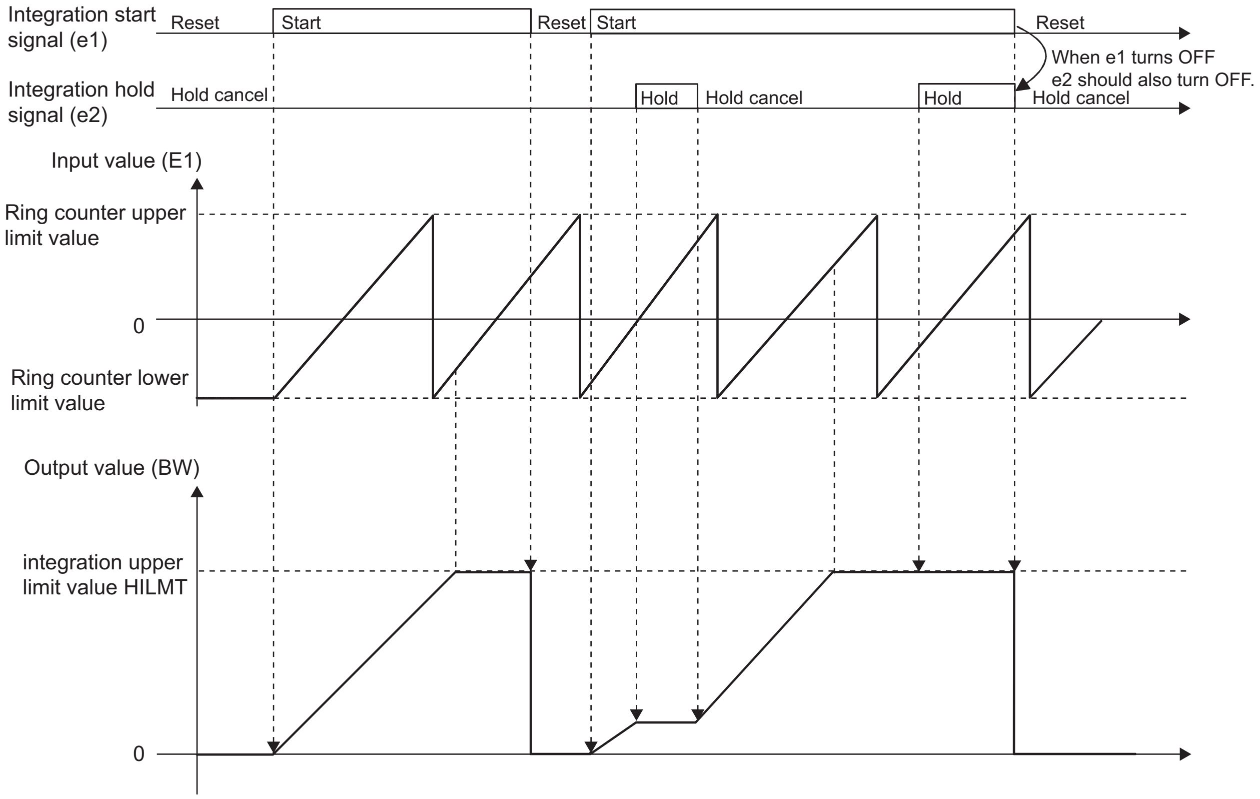
The integration upper limit value and integration pattern can be used to select whether the integrated value will be returned to 0 or retained at the upper limit value if the output value exceeds the integration upper limit value.

The integration start signal and integration hold signal can be used to start and suspend the integration of the input value.

(1) Operation performed when the integration pattern is set to "integrated value returns to 0 when the integration upper limit value is exceeded"


Integration start signal (e1)
Reset
Start
Reset
Start
Reset
When e1 turns OFF e2 should also turn OFF.
Integration hold signal (e2)
Hold cancel
Hold
Hold cancel
Hold
Hold cancel
Input value (E1)
Ring counter upper limit value
0
Ring counter lower limit value
Output value (BW)
Integration upper limit value HILMT
0

(2) Operation performed when the integration pattern is set to " integrated value is retained at the integration upper limit value when the upper limit value is exceeded"


Integration start signal (e1)
Reset
Start
Reset
Start
Reset
When e1 turns OFF e2 should also turn OFF.
Integration hold signal (e2)
Hold cancel
Hold
Hold cancel
Hold
Hold cancel
Input value (E1)
Ring counter upper limit value
0
Ring counter lower limit value
Output value (BW)
integration upper limit value HILMT
0

Set Data

(1) Data specified in S.PSUM instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value	Use the ring counter of 16 bits or more. • 16-bit ring counter $00000000_H \rightarrow 0000FFFF_H \rightarrow 00000000_H$ • 24-bit ring counter $00000000_H \rightarrow 00FFFFFF_H \rightarrow 00000000_H$ • 32-bit ring counter $00000000_H \rightarrow FFFFFFFF_H \rightarrow 00000000_H$ Set 32767 ($7FFF_H$) or less as a pulse increment at each instruction execution.	pulse	BIN 32Bit	—	U
	+2	e	—					
		e1	Integration start signal	b15 b12 b8 b4 b0 e2 e1 Integration start signal 0: Integration stop/reset 1: Integration start	—	BIN 16Bit	—	U
		e2	Integration hold signal	Integration hold signal 0: Integration hold cancel 1: Integration hold				
Block memory	Ⓓ1+0 +1	BW1	Output value (Integer part)	(0 to 2147483647)	—	BIN 32Bit	—	S
	+2 +3	BW2	Output value (Fraction part)	(0 to 2147483647)	—	BIN 32Bit	—	S
Operation constant	Ⓢ2+0	W	Weight per pulse	1 to 999	—	BIN 16Bit	1	U
	+1	U	Unit conversion constant	1, 10, 100, 1000	—	BIN 16Bit	1	U
	+2 +3	HILMT	Integration upper limit value	1 to 2147483647	—	BIN 32Bit	2147483647	U
	+4	SUMPTN	Integration pattern	0: Returns to 0 when the integration upper limit value (HILMT) is exceeded. 1: Retains the integration upper limit value when the integration upper limit value (HILMT) is exceeded.	—	BIN 16Bit	0	U
Loop tag memory [*2]	+10 +11	SUM1	Integration value (Integer part)	(0 to 2147483647)	—	BIN 32Bit	0	S
	+12 +13	SUM2	Integration value (Fraction part)	(0 to 2147483647)	—	BIN 32Bit	0	S

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[*2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Loop tag past value memory *2 *3	Ⓓ2+116 +117	—	—	Used by the system as a work area.	—	—	—	S

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

*2 The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

*3 The applications of the loop tag past value memory are indicated below.

Specified position	Description
Ⓓ2 +116 +117	$E1_{n-1}$ (Last input value)

When control is to be started from the initial status, the data must be cleared with the sequence program.

Processing contents

(1) Upper limit check processing

In the upper limit check processing, the following operation is performed and the result of the operation is output to BB2 and PHA.

e1	e2	Input value increment (T1)
0	0	—
0	1	—
1	0	$E1 - E1_{n-1}$
1	1	—

(2) Integrated value operation processing

In the integrated value operation processing, the following processing is performed for the input value increment (T1).

e1	e2	Integration value (Integer part) (T2), Integration value (Fraction part) (T3)

0	0	T2 = 0
		T3 = 0

0	1	T2 = 0
		T3 = 0 [*1]

1	0	T4 = quotient of {(T1 × W) / U} <integer part>
		T5 = remainder of {(T1 × W) / U} <fraction part>
		T2 = SUM1 + T4 + [quotient of {(SUM2 + T5) / U}] <integer part>
		T3 = remainder of {(SUM2 + T5) / U} <fraction part>

1	1	T2 = SUM1
		T3 = SUM2

*1 At an integration stop/reset (e1 = 0), processing is performed on the assumption that integration hold is canceled (e2 = 0).

(3) Output conversion

In the output conversion, the following processing is performed for the integrated value (T2, T3).

SUMPTN	Condition	BW1, SUM1	BW2, SUM2

0	T2 ≥ HILMT	BW1 = remainder of T2 / HILMT
		SUM1 = remainder of T2 / 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

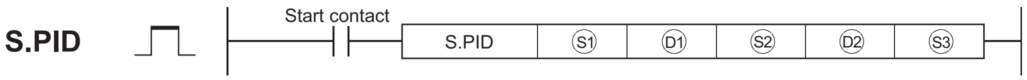
In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
4141	When an operation error occurs	—	—	○

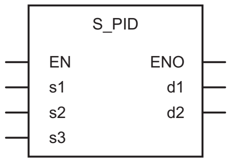
CHAPTER 9 CONTROL OPERATION INSTRUCTIONS

9.1 S.PID

Ladder diagram



Structured ladder/FBD



Structured text language

ENO:=S_PID(EN,s1,s2,s3,d1,d2);

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Real data type
	S2	: Operation constant start device	: Array of any 16-bit data (0..6)
	S3	: When set value (E2) is used: Set value start device When set value (E2) is not used: Dummy device[*1]	: Real data type
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Loop tag memory start device	: Array of any 16-bit data (0..127)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					
S3	—	○		—					

[*1] Special register SD1506 can be specified as a dummy device.

Function

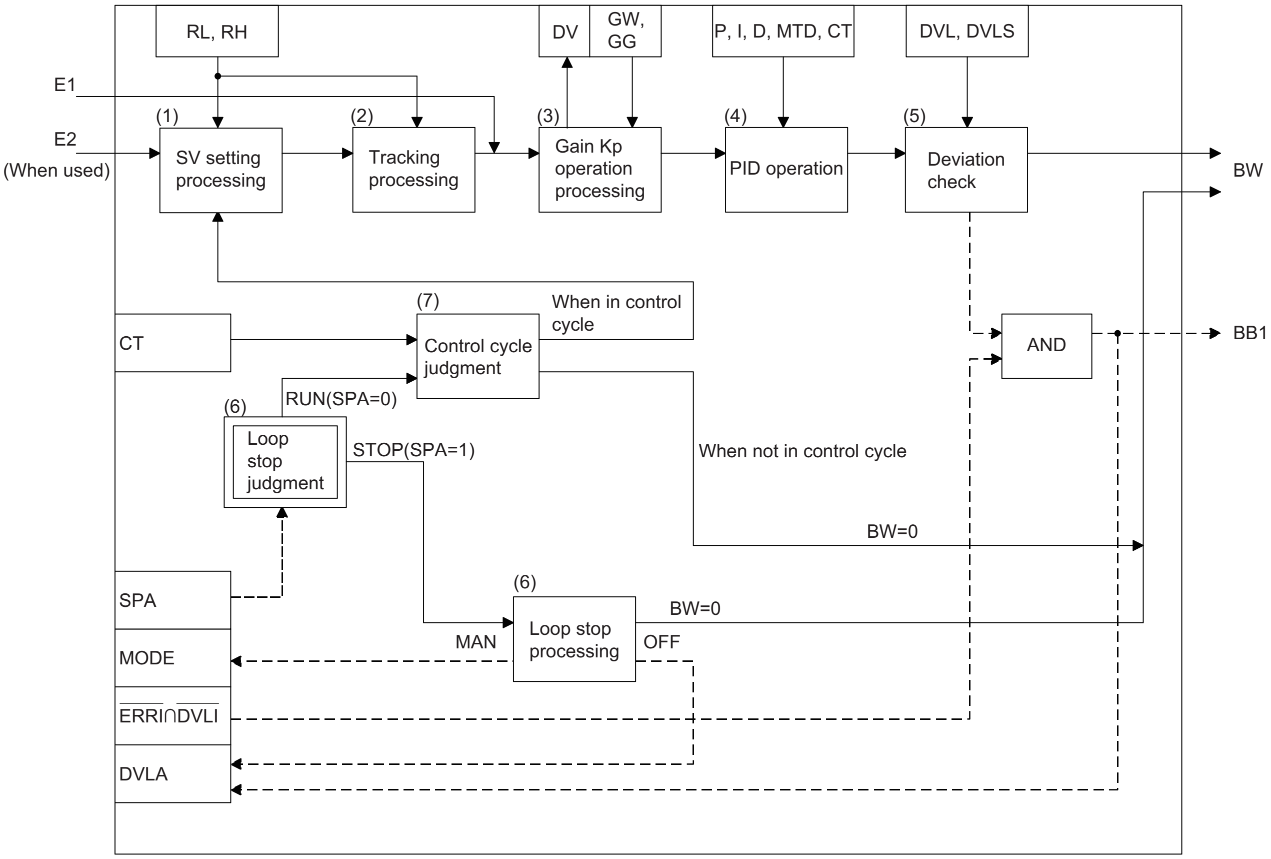
Performs PID operation when the specified control cycle is reached. (PID operation is of velocity type/process value derivative type (incomplete derivative type).)

Also performs SV setting processing, tracking processing, gain (Kp) operation processing and deviation check processing at this time.

Block diagram

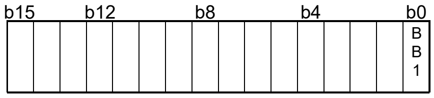
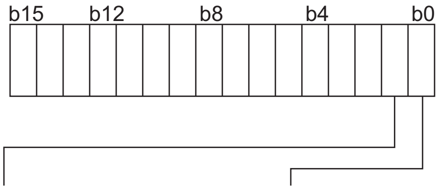
The processing block diagram of the S.PID instruction is shown below.

(The numerals (1) to (7) in the diagram indicate the order of the processing.)



Set Data

(1) Data specified in S.PID instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	S1+0 +1	E1	Input value	-999999 to 999999	—	Real number	—	U
Block memory	D1+0 +1	BW	Output value (ΔMV)	(-999999 to 999999)	—	Real number	—	S
	+2	BB	—					
		BB1	Deviation large alarm	 (0: Without alarm) (1: With alarm)	—	BIN 16bit	—	S
Operation constant	S2+0 +1	MTD	Derivative gain	0 to 999999	—	Real number	8.0	U
	+2 +3	DVLS	Deviation large alarm hysteresis	0 to 100	%	Real number	2.0	U
	+4	PN	Operation mode	0: Reverse operation 1: Forward operation	—	BIN 16bit	0	U
	+5	TRK	Tracking bit	0: Not trucked 1: Trucked	—	BIN 16bit	0	U
	+6	SVPTN	Set value pattern	0 to 3  Set value pattern[*3] 0: E2 is upper loop MV. 1: E2 is not upper loop MV. Set value used[*2] 0: E2 is used. 1: E2 is not used.	—	BIN 16bit	3	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[*2] Specify whether the set value (E2) is to be used or not.

[*3] Specify whether the MV of the upper loop is to be used or not as the set value (E2).

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Loop tag memory [*2]	(D2)+1	MODE	Operation mode	0 to FFFF$_H$ b15 b12 b8 b4 b0 [ ][ ][ ][ ][ ][CSV][CMV][CCB][CAB][CMB][CAS][AUT][MAN][LCC][LCA][LCM]	—	BIN 16bit	8$_H$	S/U
	+3	ALM	Alarm detection	0 to FFFF$_H$ b15 b12 b8 b4 b0 [ ][SPA][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][DVLA][MHA][MLA] SPA 0:Loop RUN 1:Loop STOP DVLA,MHA,MLA (0:Without alarm) (1:With alarm)	—	BIN 16bit	4000$_H$	S/U
	+4	INH	Alarm detection inhibition	0 to FFFF$_H$ b15 b12 b8 b4 b0 [ERRI][ ][TRKF][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][DVLI][MHI][MLI] TRKF (0 : Without tracking) (1 : With tracking) ERRI, DVLI, MHI, MLI 0 : Alarm enable 1 : Alarm inhibit	—	BIN 16bit	0$_H$	S/U
	+14 +15	SV	Set value	RL to RH	—	Real number	0.0	U
	+16 +17	DV	Deviation	(-110 to 110)	%	Real number	0.0	S
	+22 +23	RH	Engineering value upper limit	-999999 to 999999	—	Real number	100.0	U
	+24 +25	RL	Engineering value lower limit	-999999 to 999999	—	Real number	0.0	U
	+46 +47	CT	Control cycle	0 to 999999 Note that $\frac{CT}{\Delta T} \leqq 32767$	s	Real number	1.0	U
	+50 +51	DVL	Deviation limit value	0 to 100	%	Real number	100.0	U
	+52 +53	P	Gain	0 to 999999	—	Real number	1.0	U
	+54 +55	I	Integral constant	0 to 999999	s	Real number	10.0	U
	+56 +57	D	Derivative constant	0 to 999999	s	Real number	0.0	U
	+58 +59	GW	Gap width	0 to 100	%	Real number	0.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

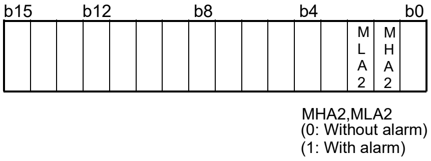
[*2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Specified position		Symbol	Name	Recommended range *1	Unit	Data format	Standard value	Set by
Loop tag memory *2	Ⓓ2+60 +61	GG	Gap gain	0 to 999999	—	Real number	1.0	U
	+62 +63	MVP	MV Inside operation value	(-999999 to 999999)	%	Real number	0.0	S
Loop tag past value memory *2 *3	Ⓓ2+96 ⋮ +116	—	—	Used by the system as a work area.	—	—	—	S
Set value *4	Ⓢ3+0 +1	E2	Set value	-10 to 110	%	Real number	0.0	U

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

*2 The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

*3 The applications of the loop tag past value memory are indicated below.

Specified position	Description
Ⓓ2+96	Control cycle counter initial preset flag
+97	Control cycle counter*5
+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} (Process value before last)
+110 +111	DV_{n-1} (Last deviation value)
+116	Alarm detection 2 (ALM2)  b15 b12 b8 b4 b0 MLA2 MHA2 MHA2,MLA2 (0: Without alarm) (1: With alarm)

When control is to be started from the initial status, the data must be cleared with the sequence program.

*4 The set value (E2) becomes valid when the set value pattern (SVPTN) is set to "E2 is used".
When using the MV of the upper loop as the set value (E2), specify the device where the manipulated value (MV) of the upper loop is set (offset + 12: MV).
When not using E2 as the set value, make sure to specify a dummy device.
(Special register SD1506 can be specified as a dummy device.)

*5 The control cycle counter rounds off the data to the nearest whole number.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) SV setting processing

Either of the following processings is performed depending on the operation mode (MODE) setting.

(a) When the operation mode (MODE) is any of CAS, CCB and CSV

1) When the set value (E2) is specified, engineering value conversion is performed with the following expression and then "(2) Tracking processing" is performed.

$$SV_n = \frac{RH - RL}{100} \times E2 + RL$$

2) When the set value (E2) is not specified, "(2) Tracking processing" is performed without the engineering value conversion being performed.

(b) When the operation mode (MODE) is any of MAN, AUT, CMV, CMB, CAB, LCM, LCA and LCC, "(2) Tracking processing" is performed.

(2) Tracking processing

(a) The set value (SV) is converted reversely from the engineering value with the following operation expression to calculate SVn'.

$$SV_n' = \frac{100}{RH - RL} \times (SV_n - RL)$$

(b) When all of the following conditions hold, tracking processing is performed.

1) The tracking bit (TRK) of the operation constant is 1.

2) The set value (E2) is used.

3) The operation mode (MODE) is any of MAN, AUT, CMV, CMB, CAB, LCM, LCA and LCC.
$E2=SV_n'$

(c) When the set value (E2) is the manipulated value (MV) of the upper loop, the tracking flag (TRKF) of the alarm detection inhibition (INH) in the upper loop turns to 1.

(3) Gain (Kp) operation processing

(a) The deviation (DV) is calculated under the following condition.

Condition	Operation expression
Forward operation(PN=1)	$DV=E1 - SV_n'$
Reverse operation(PN=0)	$DV=SV_n' - E1$

(b) The output gain (K) is calculated under the following condition.

Condition	Operation expression
When $\lvert DV \rvert \leq GW$	K=GG
When $\lvert DV \rvert > GW$	$K = 1 - \frac{(1 - GG) \times GW}{\lvert DV \rvert}$

(4) PID operation

PID operation is performed with the following operation expression.

Item		Operation expression
Bn	When forward operation (PN=1)	$B_{n-1} + \frac{M_D \times T_D}{M_D \times CT + T_D} \times \{(PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times B_{n-1}}{T_D}\}$
	When reverse operation (PN=0)	$B_{n-1} + \frac{M_D \times T_D}{M_D \times CT + T_D} \times \{-(PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times B_{n-1}}{T_D}\}$
BW (ΔMV)		$K_P \times \{(DV_n - DV_{n-1}) + \frac{CT}{T_I} \times DV_n + B_n\}$

K_P: K × Gain (P), M_D: Derivative gain (MTD)

T_I: Integral constant (I), T_D: Derivative constant (D)

In the following case, however, note that special processing will be performed.

Condition			Processing
QnPHCPU/QnPRHCPU (First 5 digits of the serial No. : 07031 or earlier)	QnPHCPU/QnPRHCPU (First 5 digits of the serial No. : 07032 or later)	QnUDPVCPU	
In either of the following cases 1, 2 1. Derivative constant (D) = 0 (T_D = 0) 2. Operation mode (MODE) is any of MAN, LCM and CMV			Bn = 0 (However, the loop tag past value memory is set.)
In any of the following cases 1, 2, 3 1. Integral constant (I) = 0 (T_I = 0) 2. When either of MHA or MLA is turned to 1 (MVP > MH) and ($\frac{CT}{T_I} \times DV_n > 0$) 3. When either of MHA or MLA is turned to 1 (MVP < ML) and ($\frac{CT}{T_I} \times DV_n < 0$)	In any of the following cases 1, 2, 3 1. Integral constant (I) = 0 (T_I = 0) 2. When either of MHA2 or MLA2 is turned to 1 (MVP > MH) and ($\frac{CT}{T_I} \times DV_n > 0$) 3. When either of MHA2 or MLA 2 is turned to 1 (MVP < ML) and ($\frac{CT}{T_I} \times DV_n < 0$)		$\frac{CT}{T_I} \times DV_n = 0$

(5) Deviation check

A deviation check is made under the following condition and the result of the check is output to DVLA of the alarm detection (ALM) and the deviation large alarm (BB1) of the block memory.

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 When DVLI or ERRI in the alarm detection inhibition (INH) is set to 1, DVLA and BB1 show 0 since the alarm is prohibited.

(6) Loop stop processing

(a) Setting 1 in SPA of the alarm detection (ALM) selects a loop stop.
A loop stop performs the following processing and terminates the S.PID instruction.

1) BW is turned to 0.
2) DVLA of the alarm detection (ALM) is turned to 0.
3) The operation mode (MODE) is changed to MAN.
4) BB1 of BB are turned to 0.

(b) Setting 0 in SPA of the alarm detection (ALM) selects a loop run.
A loop run performs "(7) Control cycle judgment".

(7) Control cycle judgment

(a) If the specified control cycle is not reached, BW (△MV) is turned to 0 and the S.PID instruction is terminated.

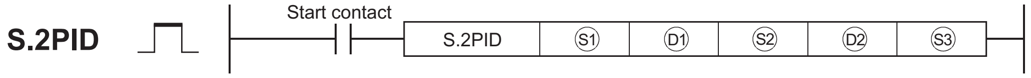
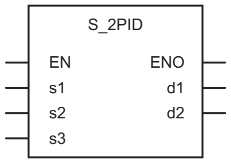
(b) When the specified control cycle is reached, "(1) SV setting processing" is performed.

Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	○	○	—
	When CT < 0 or the execution cycle (SD1500) < 0	○	○	○
4140	When the values of Ⓢ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

9.2 S.2PID

Ladder diagram	
	
Structured ladder/FBD	**Structured text language**
	ENO:=S_2PID(EN,s1,s2,s3,d1,d2);

Input argument	EN	: Execution condition	: Bit
	Ⓢ1	: Input data start device	: Real data type
	Ⓢ2	: Operation constant start device	: Array of any 16-bit data (0..6)
	Ⓢ3	: When set value (E2) is used: Set value start device When set value (E2) is not used: Dummy device[*1]	: Real data type
Output argument	ENO	: Execution result	: Bit
	Ⓓ1	: Block memory start device	: Array of any 16-bit data (0..2)
	Ⓓ2	: Loop tag memory start device	: Array of any 16-bit data (0..127)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
Ⓢ1	—	○		—					
Ⓓ1	—	○		—					
Ⓢ2	—	○		—					
Ⓓ2	—	○		—					
Ⓢ3	—	○		—					

[*1] Special register SD1506 can be specified as a dummy device.

Function

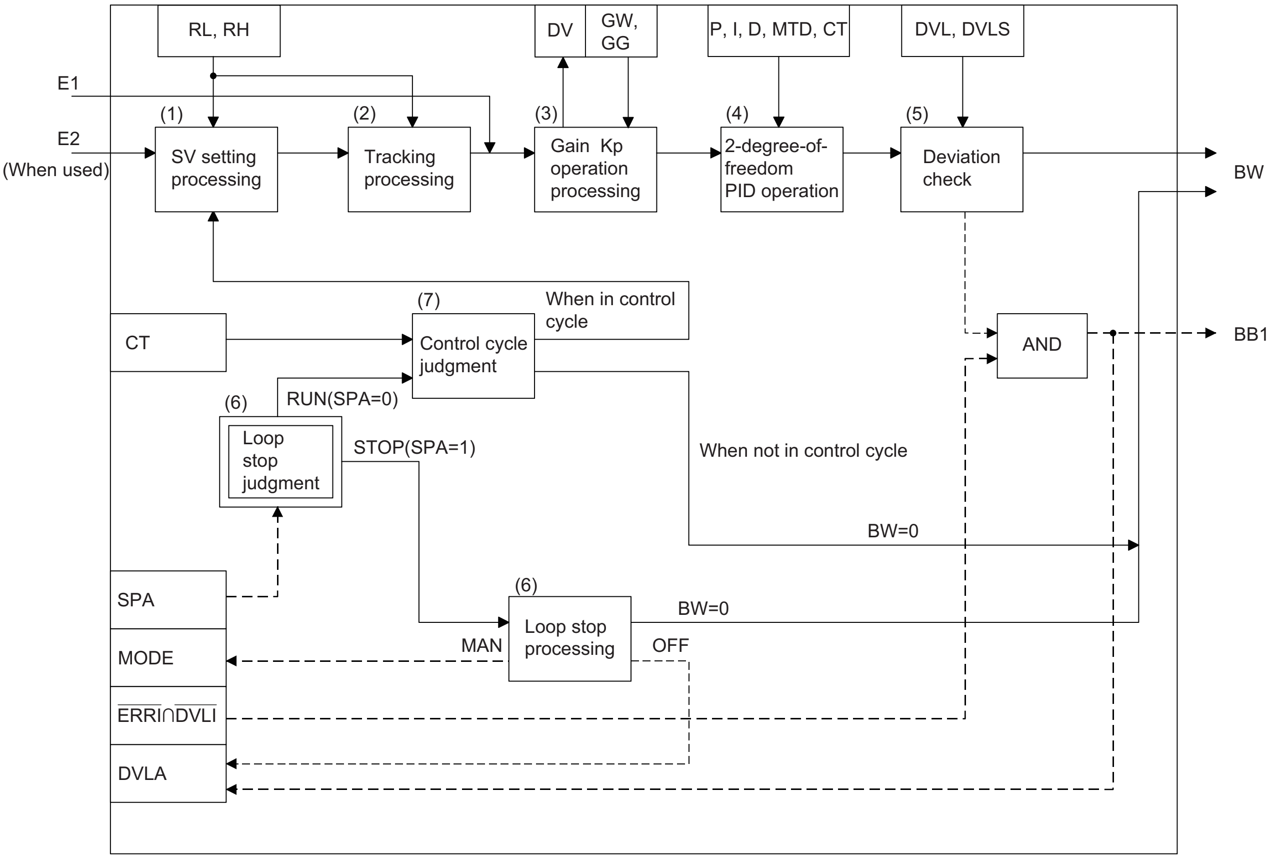
Performs 2-degree-of-freedom PID operation when the specified control cycle is reached.

Also performs SV setting processing, tracking processing, gain (Kp) operation processing and deviation check processing at this time.

Block diagram

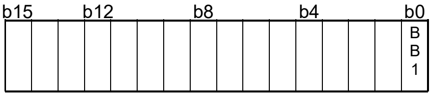
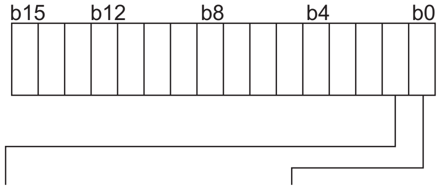
The processing block diagram of the S.2PID instruction is shown below.

(The numerals (1) to (7) in the diagram indicate the order of the processing.)



Set Data

(1) Data specified in S.2PID instruction

Specified position		Symbol	Name	Recommended range *1	Unit	Data format	Standard value	Set by
Input data	S1+0 +1	E1	Input value	-999999 to 999999	%	Real number	—	U
Block memory	D1+0 +1	BW	Output value (ΔMV)	(-999999 to 999999)	%	Real number	—	S
	+2	BB	—					
		BB1	Deviation large alarm	 (0: Without alarm) (1: With alarm)	—	BIN 16Bit	—	S
Operation constant	S2+0 +1	MTD	Derivative gain	0 to 999999	—	Real number	8.0	U
	+2 +3	DVLS	Deviation large alarm hysteresis	0 to 100	%	Real number	2.0	U
	+4	PN	Operation mode	0: Reverse operation 1: Forward operation	—	BIN 16bit	0	U
	+5	TRK	Tracking bit	0: Not tracked 1: Tracked	—	BIN 16bit	0	U
	+6	SVPTN	Set value pattern	0 to 3  Set value pattern*3 0: E2 is upper loop MV. 1: E2 is not upper loop MV. Set value used*2 0: E2 is used. 1: E2 is not used.	—	BIN 16bit	3	U

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

*2 Specify whether the set value (E2) is to be used or not.

*3 Specify whether the MV of the upper loop is to be used or not as the set value (E2).

<table>
<tr><th colspan="2">Specified position</th><th>Symbol</th><th>Name</th><th>Recommended range *1</th><th>Unit</th><th>Data format</th><th>Standard value</th><th>Set by</th></tr>
<tr><td rowspan="14">Loop tag memory *2</td><td>Ⓓ2+1</td><td>MODE</td><td>Operation mode</td><td>0 to FFFF$_H$
<table>
<tr><td>b15</td><td></td><td></td><td>b12</td><td></td><td></td><td></td><td>b8</td><td></td><td></td><td></td><td>b4</td><td></td><td></td><td></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>LCC</td><td>LCA</td><td>LCM</td></tr>
</table></td><td>—</td><td>BIN 16bit</td><td>8$_H$</td><td>S/U</td></tr>
<tr><td>+3</td><td>ALM</td><td>Alarm detection</td><td>0 to FFFF$_H$
<table>
<tr><td>b15</td><td></td><td></td><td>b12</td><td></td><td></td><td></td><td>b8</td><td></td><td></td><td></td><td>b4</td><td></td><td></td><td></td><td>b0</td></tr>
<tr><td></td><td>SPA</td><td></td><td></td><td>DMLA</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>DVLA</td><td>MHA</td><td>MLA</td></tr>
</table>
SPA
0:Loop RUN
1:Loop STOP
DVLA,DMLA,MHA,MLA
(0:Without alarm)
(1:With alarm)</td><td>—</td><td>BIN 16bit</td><td>4000$_H$</td><td>S/U</td></tr>
<tr><td>+4</td><td>INH</td><td>Alarm detection inhibition</td><td>0 to FFFF$_H$
<table>
<tr><td>b15</td><td></td><td></td><td>b12</td><td></td><td></td><td></td><td>b8</td><td></td><td></td><td></td><td>b4</td><td></td><td></td><td></td><td>b0</td></tr>
<tr><td>ERRI</td><td></td><td>TRKF</td><td></td><td>DMLI</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>DVLI</td><td>MHI</td><td>MLI</td></tr>
</table>
TRKF
(0 : Without tracking)
(1 : With tracking)
ERRI, DMLI, DVLI, MHI, MLI
0 : Alarm enable
1 : Alarm inhibit</td><td>—</td><td>BIN 16bit</td><td>0$_H$</td><td>S/U</td></tr>
<tr><td>+14
+15</td><td>SV</td><td>Set value</td><td>RL to RH</td><td>—</td><td>Real number</td><td>0.0</td><td>U</td></tr>
<tr><td>+16
+17</td><td>DV</td><td>Deviation</td><td>(-110 to 110)</td><td>%</td><td>Real number</td><td>0.0</td><td>S</td></tr>
<tr><td>+22
+23</td><td>RH</td><td>Engineering value upper limit</td><td>-999999 to 999999</td><td>—</td><td>Real number</td><td>100.0</td><td>U</td></tr>
<tr><td>+24
+25</td><td>RL</td><td>Engineering value lower limit</td><td>-999999 to 999999</td><td>—</td><td>Real number</td><td>0.0</td><td>U</td></tr>
<tr><td>+46
+47</td><td>CT</td><td>Control cycle</td><td>0 to 999999 Note that $\frac{CT}{\Delta T} \leqq 32767$</td><td>s</td><td>Real number</td><td>1.0</td><td>U</td></tr>
<tr><td>+50
+51</td><td>DVL</td><td>Deviation limit value</td><td>0 to 100</td><td>%</td><td>Real number</td><td>100.0</td><td>U</td></tr>
<tr><td>+52
+53</td><td>P</td><td>Gain</td><td>0 to 999999</td><td>—</td><td>Real number</td><td>1.0</td><td>U</td></tr>
<tr><td>+54
+55</td><td>I</td><td>Integral constant</td><td>0 to 999999</td><td>s</td><td>Real number</td><td>10.0</td><td>U</td></tr>
<tr><td>+56
+57</td><td>D</td><td>Derivative constant</td><td>0 to 999999</td><td>s</td><td>Real number</td><td>0.0</td><td>U</td></tr>
<tr><td>+58
+59</td><td>GW</td><td>Gap width</td><td>0 to 100</td><td>%</td><td>Real number</td><td>0.0</td><td>U</td></tr>
</table>

[1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

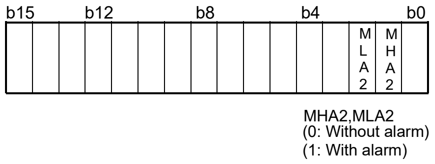
[2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Specified position		Symbol	Name	Recommended range [1]	Unit	Data format	Standard value	Set by
Loop tag memory [2]	Ⓓ2+60 +61	GG	Gap gain	0 to 999999	—	Real number	1.0	U
	+62 +63	MVP	MV inside operation value	(-999999 to 999999)	%	Real number	0.0	S
	+64 +65	α	2 degree-of-freedom parameter α [5]	0 to 1	—	Real number	0.0	U
	+66 +67	β	2 degree-of-freedom parameter β [6]	0 to 1	—	Real number	1.0	U
Loop tag past value memory [2] [3]	+96 ⋮ +116	—	—	Used by the system as a work area.	—	—	—	S
Set value [4]	Ⓢ3+0 +1	E2	Set value	-10 to 110	%	Real number	0.0	U

[1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

[3] The applications of the loop tag past value memory are indicated below.

Specified position	Description
Ⓓ2+96	Control cycle counter initial preset flag
+97	Control cycle counter [7]
+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} (Process value before last)
+110 +111	DV_{n-1} (Last deviation value)
+112 +113	DV_{n-2} (Deviation value before last)
+114 +115	D_{n-1} (Last value)
+116	Alarm detection 2 (ALM2)  b15 b12 b8 b4 b0 MLA2 MHA2 MHA2,MLA2 (0: Without alarm) (1: With alarm)

When control is to be started from the initial status, the data must be cleared with the sequence program.

[4] The set value (E2) becomes valid when the set value pattern (SVPTN) is set to "E2 is used".
When using the MV of the upper loop as the set value (E2), specify the device where the manipulated value (MV) of the upper loop is set (offset + 12: MV).
When not using E2 as the set value, make sure to specify a dummy device.
(Special register SD1506 can be specified as a dummy device.)

[5] Increasing α decreases the manipulated value variation relative to the set value change.
(It will take time to stabilize.)
Decreasing α increases the manipulated value variation relative to the set value change.
However, since a compensation operation will be stronger, hunting may become greater.

[6] Increasing β decreases the effect of derivative on the set value change.
Decreasing β increases the effect of derivative on the set value change.

*7 The control cycle counter rounds off the data to the nearest whole number.

(2) Execution cycle (∆T)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) SV setting processing

Either of the following processings is performed depending on the operation mode (MODE) setting.

(a) When the operation mode (MODE) is any of CAS, CCB and CSV

1) When the set value (E2) is specified, engineering value conversion is performed with the following expression and then "(2) Tracking processing" is performed.

$$SV_n=\frac{RH-RL}{100}\times E2+RL$$

2) When the set value (E2) is not specified, "(2) Tracking processing" is performed without the engineering value conversion being performed.

(b) When the operation mode (MODE) is any of MAN, AUT, CMV, CMB, CAB, LCM, LCA and LCC, "(2) Tracking processing" is performed.

(2) Tracking processing

(a) The set value (SV) is converted reversely from the engineering value with the following operation expression to calculate SVn'.

$$SV_n'=\frac{100}{RH-RL}\times(SV_n-RL)$$

(b) When all of the following conditions hold, tracking processing is performed.

1) The tracking bit (TRK) of the operation constant is 1.

2) The set value (E2) is used.

3) The operation mode (MODE) is any of MAN, AUT, CMV, CMB, CAB, LCM, LCA and LCC.
E2=SVn'

(c) When the set value (E2) is the manipulated value (MV) of the upper loop, the tracking flag (TRKF) of the alarm detection inhibition (INH) in the upper loop turns to 1.

(3) Gain (Kp) operation processing

(a) The deviation (DV) is calculated under the following condition.

Condition	Operation expression
Forward operation (PN=1)	DV=E1 - SVn'
Reverse operation (PN=0)	DV=SVn' - E1

(b) The output gain (K) is calculated under the following condition.

Condition	Operation expression
When |DV| ≤ GW	K=GG
When |DV| > GW	$K=1-\frac{(1-GG)\times GW}{\|DV\|}$

(4) 2-degree-of-freedom PID operation

2-degree-of-freedom PID operation is performed with the following operation expression.

Item		Operation expression
B_n		$B_{n-1} + \frac{M_D \times T_D}{M_D \times CT + T_D} \times \{(DV_n - 2DV_{n-1} + DV_{n-2}) - \frac{CT \times B_{n-1}}{T_D}\}$
C_n	When forward operation (PN=1)	$PV_n - PV_{n-1}$
	When reverse operation (PN=0)	$-(PV_n - PV_{n-1})$
D_n	When forward operation (PN=1)	$D_{n-1} + \frac{M_D \times T_D}{M_D \times CT + T_D} \times \{(PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times D_{n-1}}{T_D}\}$
	When reverse operation (PN=0)	$D_{n-1} + \frac{M_D \times T_D}{M_D \times CT + T_D} \times \{-(PV_n - 2PV_{n-1} + PV_{n-2}) - \frac{CT \times D_{n-1}}{T_D}\}$
BW (ΔMV)		s

K_P: K × Gain (P), M_D: Derivative gain (MTD)

T_I: Integral constant (I), T_D: Derivative constant (D)

In the following case, however, note that special processing will be performed.

Condition			Processing
QnPHCPU/QnPRHCPU (First 5 digits of the serial No. : 07031 or earlier)	**QnPHCPU/QnPRHCPU (First 5 digits of the serial No. : 07032 or later)**	**QnUDPVCPU**	
In either of the following cases 1, 2 1. Derivative constant (D) = 0 (T_D = 0) 2. Operation mode (MODE) is any of MAN, LCM and CMV			$B_n = 0, D_n = 0$ (However, the loop tag past value memory is set.)
In any of the following cases 1, 2, 3 1. Integral constant (I) = 0 (T_I = 0) 2. When either of MHA or MLA is turned to 1 (MVP > MH) and ($\frac{CT}{T_I} \times DV_n > 0$) 3. When either of MHA or MLA is turned to 1 (MVP < ML) and ($\frac{CT}{T_I} \times DV_n < 0$)	In any of the following cases 1, 2, 3 1. Integral constant (I) = 0 (T_I = 0) 2. When either of MHA2 or MLA2 is turned to 1 (MVP > MH) and ($\frac{CT}{T_I} \times DV_n > 0$) 3. When either of MHA2 or MLA 2 is turned to 1 (MVP < ML) and ($\frac{CT}{T_I} \times DV_n < 0$)		$\frac{CT}{T_I} \times DV_n = 0$

(5) Deviation check

A deviation check is made under the following condition and the result of the check is output to DVLA of the alarm detection (ALM) and the deviation large alarm (BB1) of the block memory.

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 When DVLI or ERRI in the alarm detection inhibition (INH) is set to 1, DVLA and BB1 show 0 since the alarm is prohibited.

(6) Loop stop processing

(a) Setting 1 in SPA of the alarm detection (ALM) selects a loop stop.
A loop stop performs the following processing and terminates the S.2PID instruction.

1) BW is turned to 0.

2) DVLA of the alarm detection (ALM) is turned to 0.

3) The operation mode (MODE) is changed to MAN.

4) BB1 of BB are turned to 0.

(b) Setting 0 in SPA of the alarm detection (ALM) selects a loop run.
A loop run performs "(7) Control cycle judgment".

(7) Control cycle judgment

(a) If the specified control cycle is not reached, BW (△MV) is turned to 0 and the S.2PID instruction is terminated.

(b) When the specified control cycle is reached, "(1) SV setting processing" is performed.

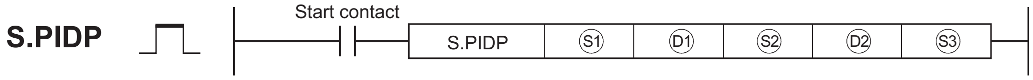
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

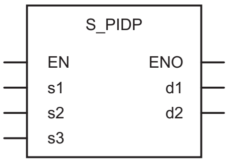
Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of S1, S2, D2 are either a non-numeric or non-normalized number	○	○	—
	When CT < 0 or the execution cycle (SD1500) < 0	○	○	○
4140	When the values of S1, S2, D2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

9.3 S.PIDP

Ladder diagram



Structured ladder/FBD



Structured text language

ENO:=S_PIDP(EN,s1,s2,s3,d1,d2);

Input argument	EN	: Execution condition	: Bit
	(S1)	: Input data start device	: Real data type
	(S2)	: Operation constant start device	: Array of any 16-bit data (0..10)
	(S3)	: When set value (E2) is used: Set value start device When set value (E2) is not used: Dummy device[*1]	: Real data type
Output argument	ENO	: Execution result	: Bit
	(D1)	: Block memory start device	: Array of any 16-bit data (0..2)
	(D2)	: Loop tag memory start device	: Array of any 16-bit data (0..127)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
(S1)	—	○		—					
(D1)	—	○		—					
(S2)	—	○		—					
(D2)	—	○		—					
(S3)	—	○		—					

[*1] Special register SD1506 can be specified as a dummy device.

Function

Performs position type PID operation when the specified control cycle is reached.

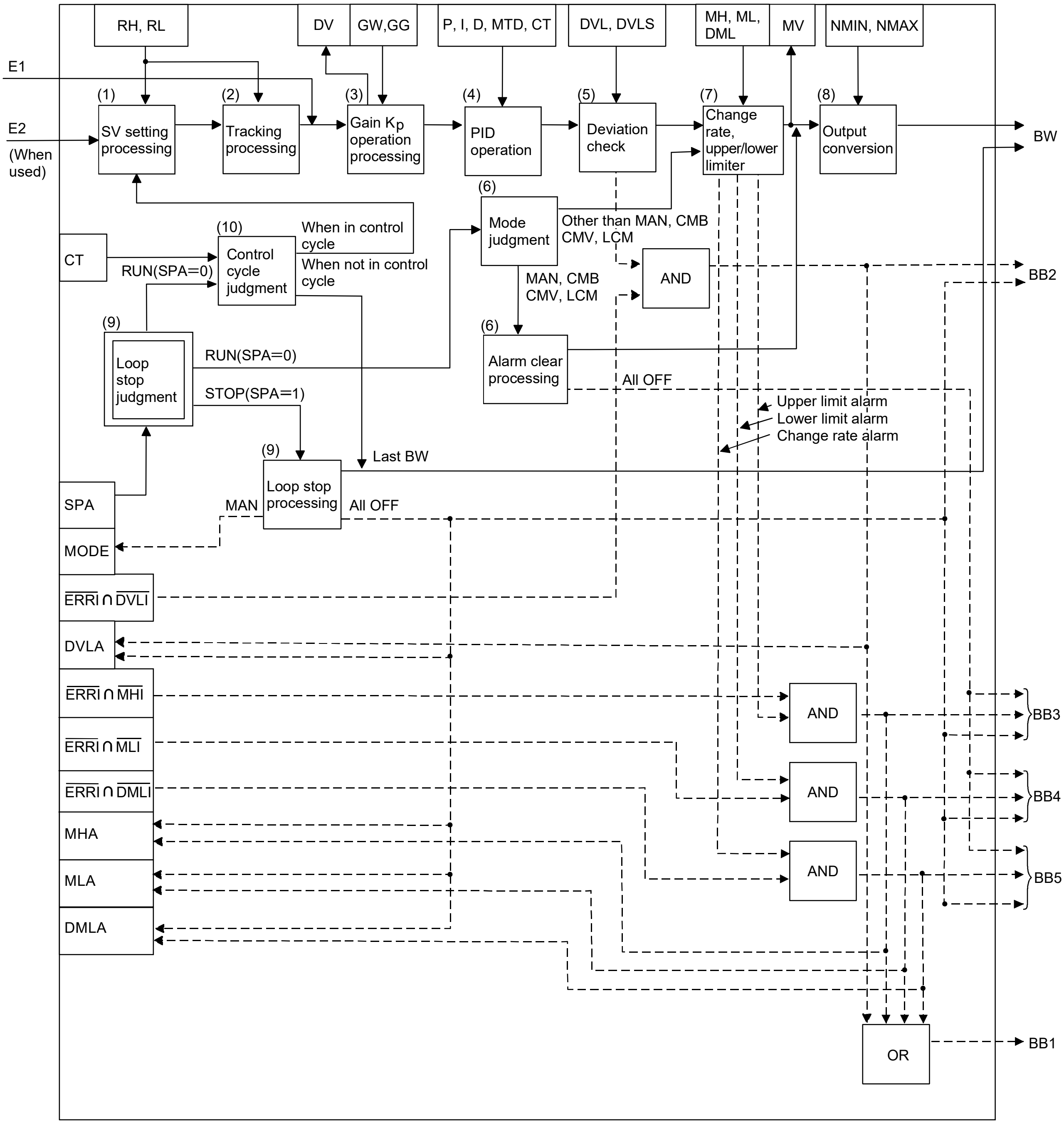
Also performs SV setting processing, tracking processing, gain (Kp) operation processing, deviation check processing and operation mode (MODE) judgment at this time.

Performs change rate, upper/lower limiter and output processings or alarm clear processing and output conversion according to the result of the judgment.

Block diagram

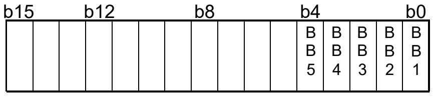
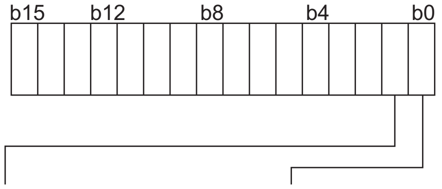
The processing block diagram of the S.PIDP instruction is shown below.

(The numerals (1) to (10) in the diagram indicate the order of the processing.)



Set Data

(1) Data specified in S.PIDP instruction

Specified position		Symbol	Name	Recommended range [1]	Unit	Data format	Standard value	Set by
Input data	S1+0 +1	E1	Input value	-999999 to 999999	%	Real number	—	U
Block memory	D1+0 +1	BW	Output value	(-999999 to 999999)	—	Real number	—	S
	+2	BB	—					
		BB1	Alarm	 b15 b12 b8 b4 b0 BB5 BB4 BB3 BB2 BB1 (0: Without alarm) (1: With alarm)	—	BIN 16bit	—	S
		BB2	Deviation large alarm					
		BB3	Output upper limit alarm					
		BB4	Output lower limit alarm					
		BB5	Output change rate alarm					
Operation constant	S2+0 +1	MTD	Derivative gain	0 to 999999	—	Real number	8.0	U
	+2 +3	DVLS	Deviation large alarm hysteresis	0 to 100	%	Real number	2.0	U
	+4	PN	Operation mode	0: Reverse operation 1: Forward operation	—	BIN 16bit	0	U
	+5	TRK	Tracking bit	0: Not trucked 1: Trucked	—	BIN 16bit	0	U
	+6	SVPTN	Set value pattern	0 to 3  Set value pattern[3] 0: E2 is upper loop MV. 1: E2 is not upper loop MV. Set value used[2] 0: E2 is used. 1: E2 is not used.	—	BIN 16bit	3	U
	+7 +8	NMAX	Output conversion upper limit	-999999 to 999999	—	Real number	100.0	U
	+9 +10	NMIN	Output conversion lower limit	-999999 to 999999	—	Real number	0.0	U

[1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[2] Specify whether the set value (E2) is to be used or not.

[3] Specify whether the MV of the upper loop is to be used or not as the set value (E2).

<table>
<tr><th colspan="2">Specified position</th><th>Symbol</th><th>Name</th><th>Recommended range [*1]</th><th>Unit</th><th>Data format</th><th>Standard value</th><th>Set by</th></tr>
<tr><td rowspan="13">Loop tag memory [*2]</td><td>Ⓓ2+1</td><td>MODE</td><td>Operation mode</td><td>0 to FFFF_H
<table><tr><td>b15</td><td></td><td></td><td>b12</td><td></td><td></td><td></td><td>b8</td><td></td><td></td><td></td><td>b4</td><td></td><td></td><td></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>LCC</td><td>LCA</td><td>LCM</td></tr></table></td><td>—</td><td>BIN
16bit</td><td>8_H</td><td>S/U</td></tr>
<tr><td>+3</td><td>ALM</td><td>Alarm detection</td><td>0 to FFFF_H
<table><tr><td>b15</td><td></td><td></td><td>b12</td><td></td><td></td><td></td><td>b8</td><td></td><td></td><td></td><td>b4</td><td></td><td></td><td></td><td>b0</td></tr><tr><td></td><td>SPA</td><td></td><td></td><td>DMLA</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>DVLA</td><td>MHA</td><td>MLA</td></tr></table>SPA
0:Loop RUN
1:Loop STOP
DVLA,DMLA,MHA,MLA
(0:Without alarm)
(1:With alarm)</td><td>—</td><td>BIN
16bit</td><td>4000_H</td><td>S/U</td></tr>
<tr><td>+4</td><td>INH</td><td>Alarm detection inhibition</td><td>00 to FFFF_H
<table><tr><td>b15</td><td></td><td></td><td>b12</td><td></td><td></td><td></td><td>b8</td><td></td><td></td><td></td><td>b4</td><td></td><td></td><td></td><td>b0</td></tr><tr><td>ERRI</td><td></td><td>TRKF</td><td></td><td>DMLI</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>DVLI</td><td>MHI</td><td>MLI</td></tr></table>TRKF
(0 : Without tracking)
(1 : With tracking)
ERRI, DMLI, DVLI, MHI, MLI
0 : Alarm enable
1 : Alarm inhibit</td><td>—</td><td>BIN
16bit</td><td>0_H</td><td>S/U</td></tr>
<tr><td>+12
+13</td><td>MV</td><td>Manipulated value</td><td>-10 to 110</td><td>%</td><td>Real number</td><td>0.0</td><td>S/U</td></tr>
<tr><td>+14
+15</td><td>SV</td><td>Set value</td><td>RL to RH</td><td>—</td><td>Real number</td><td>0.0</td><td>U</td></tr>
<tr><td>+16
+17</td><td>DV</td><td>Deviation</td><td>(-110 to 110)</td><td>%</td><td>Real number</td><td>0.0</td><td>S</td></tr>
<tr><td>+18
+19</td><td>MH</td><td>Output upper limit value</td><td>-10 to 110</td><td>%</td><td>Real number</td><td>100.0</td><td>U</td></tr>
<tr><td>+20
+21</td><td>ML</td><td>Output lower limit value</td><td>-10 to 110</td><td>%</td><td>Real number</td><td>0.0</td><td>U</td></tr>
<tr><td>+22
+23</td><td>RH</td><td>Engineering value upper limit</td><td>-999999 to 999999</td><td>—</td><td>Real number</td><td>100.0</td><td>U</td></tr>
<tr><td>+24
+25</td><td>RL</td><td>Engineering value lower limit</td><td>-999999 to 999999</td><td>—</td><td>Real number</td><td>0.0</td><td>U</td></tr>
<tr><td>+46
+47</td><td>CT</td><td>Control cycle</td><td>0 to 999999 Note that $\frac{CT}{\Delta T} \leqq 32767$</td><td>s</td><td>Real number</td><td>1.0</td><td>U</td></tr>
<tr><td>+48
+49</td><td>DML</td><td>Output change rate limit value</td><td>0 to 100</td><td>%</td><td>Real number</td><td>100.0</td><td>U</td></tr>
<tr><td>+50
+51</td><td>DVL</td><td>Deviation limit value</td><td>0 to 100</td><td>%</td><td>Real number</td><td>100.0</td><td>U</td></tr>
</table>

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

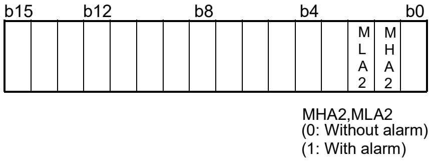
[*2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Specified position		Symbol	Name	Recommended range [1]	Unit	Data format	Standard value	Set by
Loop tag memory [2]	Ⓓ2+52 +53	P	Gain	0 to 999999	—	Real number	1.0	U
	+54 +55	I	Integral constant	0 to 999999	s	Real number	10.0	U
	+56 +57	D	Derivative constant	0 to 999999	s	Real number	0.0	U
	+58 +59	GW	Gap width	0 to 100	%	Real number	0.0	U
	+60 +61	GG	Gap gain	0 to 999999	—	Real number	1.0	U
Loop tag past value memory [2] [3]	Ⓓ2+96 ⋮ +116	—	—	Used by the system as a work area.	—	—	—	S
Set value [4]	Ⓢ3+0 +1	E2	Set value	-10 to 110	%	Real number	0.0	U

[1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

[3] The applications of the loop tag past value memory are indicated below.

Specified position	Description
Ⓓ2+96	Control cycle counter initial preset flag
+97	Control cycle counter [5]
+100 +101	I_{n-1} (Last value)
+102 +103	B_{n-1} (Last value)
+104 +105	PV_n (Process value)
+106 +107	PV_{n-1} (Last process value)
+116	Alarm detection 2 (ALM2)  b15 b12 b8 b4 b0 MLA2 MHA2 MHA2,MLA2 (0: Without alarm) (1: With alarm)

When control is to be started from the initial status, the data must be cleared with the sequence program.

[4] The set value (E2) becomes valid when the set value pattern (SVPTN) is set to "E2 is used".
When using the MV of the upper loop as the set value (E2), specify the device where the manipulated value (MV) of the upper loop is set (offset + 12: MV).
When not using E2 as the set value, make sure to specify a dummy device.
(Special register SD1506 can be specified as a dummy device.)

[5] The control cycle counter rounds off the data to the nearest whole number.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) SV setting processing

Either of the following processings is performed depending on the operation mode (MODE) setting.

(a) When the operation mode (MODE) is any of CAS, CCB and CSV

1) When the set value (E2) is specified, engineering value conversion is performed with the following expression and then "(2) Tracking processing" is performed.

$$SV_n = \frac{RH-RL}{100} \times E2 + RL$$

2) When the set value (E2) is not specified, "(2) Tracking processing" is performed without the engineering value conversion being performed.

(b) When the operation mode (MODE) is any of MAN, AUT, CMV, CMB, CAB, LCM, LCA and LCC, "(2) Tracking processing" is performed.

(2) Tracking processing

(a) The set value (SV) is converted reversely from the engineering value with the following operation expression to calculate SVn'.

$$SV_n' = \frac{100}{RH-RL} \times (SV_n - RL)$$

(b) When all of the following conditions hold, tracking processing is performed.

1) The tracking bit (TRK) of the operation constant is 1.

2) The set value (E2) is used.

3) The operation mode (MODE) is any of MAN, AUT, CMV, CMB, CAB, LCM, LCA and LCC.
E2=SVn'

(c) When the set value (E2) is the manipulated value (MV) of the upper loop, the tracking flag (TRKF) of the alarm detection inhibition (INH) in the upper loop turns to 1.

(3) Gain (Kp) operation processing

(a) The deviation (DV) is calculated under the following condition.

Condition	Operation expression
Forward operation (PN=1)	DV=E1 - SVn'
Reverse operation (PN=0)	DV=SVn' - E1

(b) The output gain (K) is calculated under the following condition.

Condition	Operation expression
When |DV| ≤ GW	K=GG
When |DV| > GW	$K = 1 - \frac{(1-GG) \times GW}{\|DV\|}$

(4) PID operation

PID operation is performed with the following operation expression.

Item		Operation expression
Bn	When forward operation (PN = 1)	$B_{n-1}+\frac{M_D\times T_D}{M_D\times CT+T_D}\times\{(PV_n-PV_{n-1})-\frac{CT\times B_{n-1}}{T_D}\}$
	When reverse operation (PN = 0)	$B_{n-1}+\frac{M_D\times T_D}{M_D\times CT+T_D}\times\{-(PV_n-PV_{n-1})-\frac{CT\times B_{n-1}}{T_D}\}$
In		$I_{n-1}+\frac{CT}{T_I}\times DV_n$
T		Kp×(DVn + In + Bn)

K_P: K × Gain (P), M_D: Derivative gain (MTD)

T_I: Integral constant (I), T_D: Derivative constant (D)

In the following case, however, note that special processing will be performed.

Condition			Processing
QnPHCPU (First 5 digits of the serial No. : 07031 or earlier)	QnPHCPU (First 5 digits of the serial No. : 07032 or later)	QnUDPVCPU	
In either of the following cases 1, 2 1. Derivative constant (D) = 0 (T_D = 0) 2. Operation mode (MODE) is any of MAN, LCM and CMV			B_n = 0 (However, the loop tag past value memory is set.)
In any of the following cases 1, 2, 3 1. Integral constant (I) = 0 (T_I = 0) 2. When MHA is turned to 1 $\frac{CT}{T_I}\times DV_n>0$ 3. When MLA is turned to 1 $\frac{CT}{T_I}\times DV_n<0$	In any of the following cases 1, 2, 3, 4 1. Integral constant (I) = 0 (T_I = 0) 2. When MHA2 is turned to 1 $\frac{CT}{T_I}\times DV_n>0$ 3. When MLA2 is turned to 1 $\frac{CT}{T_I}\times DV_n<0$ 4. When operating mode (MODE) is any of MAN, LCM, and CMV.		$\frac{CT}{T_I}\times DV_n=0$
—	All the following conditions 1, 2, 3 are satisfied 1. When b0 of SD1508 is turned to 1 2. When tracking flag (TRKF) in alarm detection inhibition (INH) is turned to 1. 3. When operating mode (MODE) is other than MAN, LCM, and CMV.		$I_{n-1}=\frac{MV}{K_p}-(DV_n+B_n)$ TRKF=0

(5) Deviation check

A deviation check is made under the following condition and the result of the check is output to DVLA of the alarm detection (ALM) and the deviation large alarm (BB2) of the block memory.

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] When DVLI or ERRI in the alarm detection inhibition (INH) is set to 1, DVLA and BB2 show 0 since the alarm is prohibited.

(6) Mode judgment

Either of the following processings is performed depending on the operation mode (MODE) setting.

(a) When the operation mode (MODE) is any of MAN, CMB, CMV and LCM (alarm clear processing)

1) MHA, MLA and DMLA of the alarm detection (ALM) are turned to 0.

2) MAH2 and MLA2 of the alarm detection 2 (ALM2) are turned to 0.

3) BB3 to BB5 of BB are turned to 0.

4) Data of BB2 is transferred to BB1 of BB. (BB1 = BB2)

5) "(8) Output conversion processing" is performed and the S.PIDP instruction is terminated.

(b) When the operation mode (MODE) is any of AUT, CAB, CAS, CCB, CSV, LCA and LCC, "(7) Change rate, upper/lower limiter" is executed.

(7) Change rate, upper/lower limiter

The change rate and upper/lower limits are checked for the input value (E1), and the data after the limiter processing and an alarm are output.

(a) The change rate limiter performs the following operation and outputs the result of the operation to BB5 and DMLA.

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 When DMLI or ERRI in the alarm detection inhibition (INH) is set to 1, DMLA and BB5 show 0 since the alarm is prohibited.

(b) The upper/lower limiter performs the following operation and outputs the result of the operation to BB3, BB4, MHA, MLA, MHA2 and MLA2.

Condition	BB4, MLA, MLA2	BB3, MHA, MHA2	MV
T1 > MH	0	1 [*2]	MH
T1 < ML	1 [*3]	0	ML
$ML \leq T1 \leq MH$	0	0	T1

*2 When MHI or ERRI in the alarm detection inhibition (INH) is set to 1, MHA and BB3 show 0 since the alarm is prohibited. However, even if MHI and/or ERRI in the alarm detection inhibition (INH) is set to 1, MHA2 holds 1.

*3 When MLI or ERRI in the alarm detection inhibition (INH) is set to 1, MLA and BB4 show 0 since the alarm is prohibited. However, even if MLI and/or ERRI in the alarm detection inhibition (INH) is set to 1, MLA2 holds 1.

(8) Output conversion

In the output conversion, the output value is calculated from the following formula.

$$BW = \frac{NMAX - NMIN}{100} \times MV + NMIN$$

(9) Loop stop processing

(a) Setting 1 in SPA of the alarm detection (ALM) selects a loop stop.

A loop stop performs the following processing and terminates the S.PIDP instruction.

1) BW retains the last value.

2) DVLA, MHA, MLA and DMLA of the alarm detection (ALM) are turned to 0.

3) MHA2 and MLA2 of the alarm detection 2 (ALM2) are turned to 0.

4) The operation mode (MODE) is changed to MAN.

5) BB1 to BB5 of BB are turned to 0.

(b) Setting 0 in SPA of the alarm detection (ALM) selects a loop run.

A loop run performs "(10) Control cycle judgement".

(10) Control cycle judgment

(a) When the specified control cycle is not reached, " (6) mode judgement" as T = MV is performed.

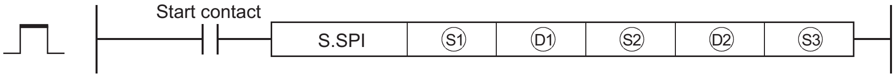
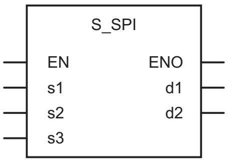
(b) When the specified control cycle is reached, "(1) SV setting processing" is performed.

Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored into SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	○	○	—
	When CT < 0 or the execution cycle (SD1500) < 0	○	○	○
4140	When the values of Ⓢ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

9.4 S.SPI

Ladder diagram	
S.SPI 	
Structured ladder/FBD	**Structured text language**
	ENO:=S_SPI(EN,s1,s2,s3,d1,d2);

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Real data type
	S2	: Operation constant start device	: Array of any 16-bit data (0..4)
	S3	: When set value (E2) is used: Set value start device When set value (E2) is not used: Dummy device*1	: Real data type
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Loop tag memory start device	: Array of any 16-bit data (0..127)

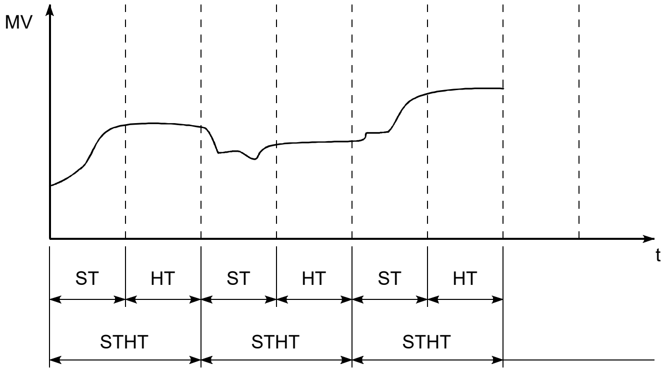
Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					
S3	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

Function

Performs normal PI operation during operating time (ST).

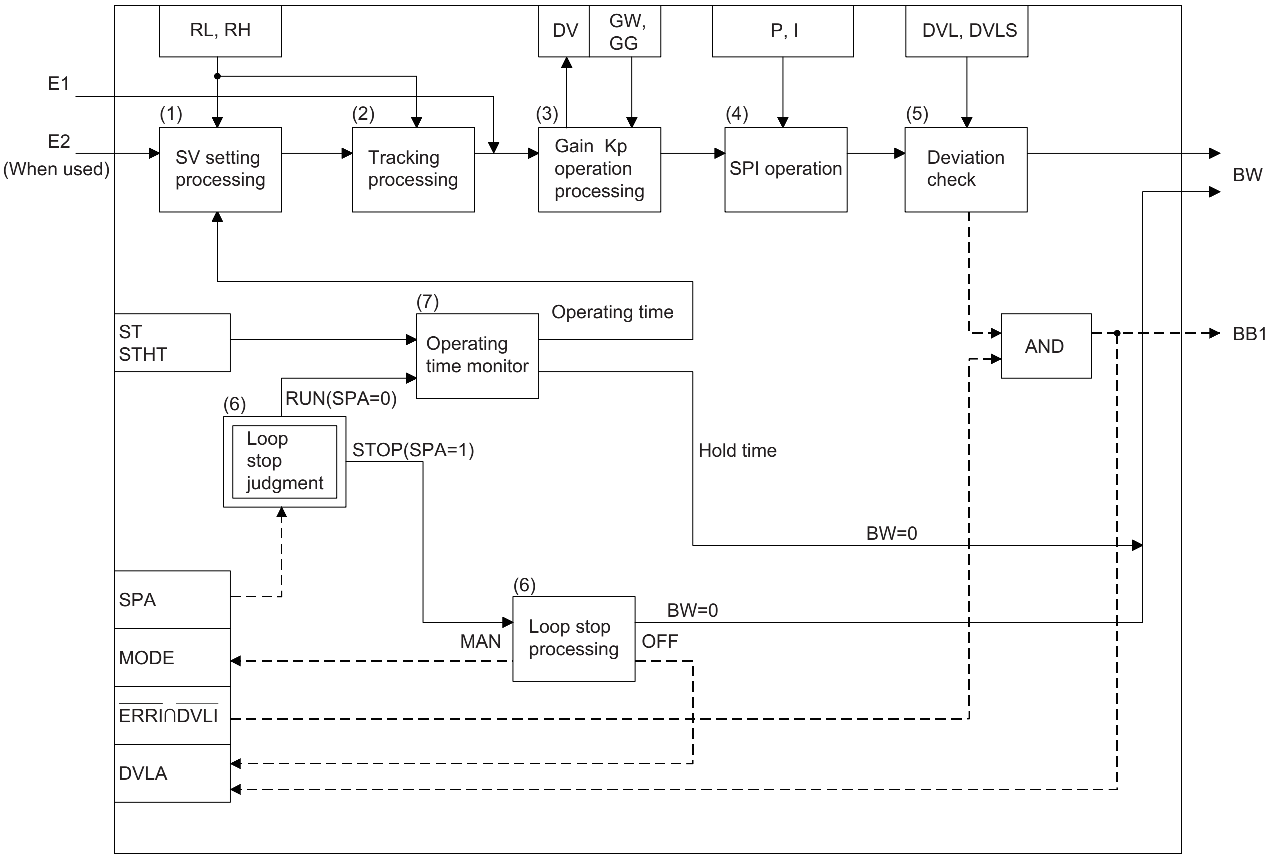
Judges between operating time (ST) or hold time (HT), and if it is the operating time, performs SV setting processing, tracking processing, gain (Kp) operation processing, SPI operation and deviation check.



Block diagram

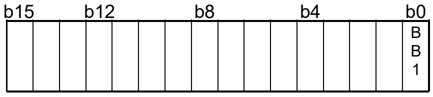
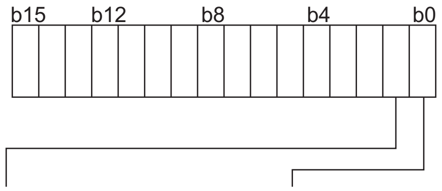
The processing block diagram of the S.SPI instruction is shown below.

(The numerals (1) to (7) in the diagram indicate the order of the processing.)



Set Data

(1) Data specified in S.SPI instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value	-999999 to 999999	%	Real number	—	U
Block memory	Ⓓ1+0 +1	BW	Output value (ΔMV)	(-999999 to 999999)	%	Real number	—	S
	+2	BB	—					
		BB1	Deviation large alarm	 (0: Without alarm) (1: With alarm)	—	BIN 16bit	—	S
Operation constant	Ⓢ2+0 +1	DVLS	Deviation large alarm hysteresis	0 to 100	%	Real number	2.0	U
	+2	PN	Operation mode	0: Reverse operation 1: Forward operation	—	BIN 16bit	0	U
	+3	TRK	Tracking bit	0: Not trucked 1: Trucked	—	BIN 16bit	0	U
	+4	SVPTN	Set value pattern	0 to 3  Set value pattern[*3] 0: E2 is upper loop MV. 1: E2 is not upper loop MV. Set value used[*2] 0: E2 is used. 1: E2 is not used.	—	BIN 16bit	3	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[*2] Specify whether the set value (E2) is to be used or not.

[*3] Specify whether the MV of the upper loop is to be used or not as the set value (E2).

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Loop tag memory [*2]	Ⓓ2+1	MODE	Operation mode	0 to FFFF$_H$ b15 b12 b8 b4 b0 □ □ □ □ □ CSV CMV CCB CAB CMB CAS AUT MAN LCC LCA LCM	—	BIN 16Bit	8$_H$	S/U
	+3	ALM	Alarm detection	0 to FFFF$_H$ b15 b12 b8 b4 b0 □ SPA □ □ □ □ □ □ □ □ □ □ □ DVLA MHA MLA SPA 0:Loop RUN 1:Loop STOP DVLA,MHA,MLA (0:Without alarm) (1:With alarm)	—	BIN 16Bit	4000$_H$	S/U
	+4	INH	Alarm detection inhibition	00 to FFFF$_H$ b15 b12 b8 b4 b0 ERRI □ TRKF □ □ □ □ □ □ □ □ □ □ DVLI MHI MLI TRKF (0 : Without tracking) (1 : With tracking) ERRI, DVLI, MHI, MLI 0 : Alarm enable 1 : Alarm inhibit	—	BIN 16Bit	0$_H$	S/U
	+14 +15	SV	Set value	RL to RH	—	Real number	0.0	U
	+16 +17	DV	Deviation	(-110 to 110)	%	Real number	0.0	S
	+22 +23	RH	Engineering value upper limit	-999999 to 999999	—	Real number	100.0	U
	+24 +25	RL	Engineering value lower limit	-999999 to 999999	—	Real number	0.0	U
	+46 +47	ST	Operating time	0 to 999999 Note that $\frac{ST}{\Delta T} \leqq 32767$	s	Real number	0.0	U
	+50 +51	DVL	Deviation limit value	0 to 100	%	Real number	100.0	U
	+52 +53	P	Gain	0 to 999999	—	Real number	1.0	U
	+54 +55	I	Integral constant	0 to 999999	s	Real number	10.0	U
	+56 +57	STHT	Sample cycle	0 to 999999 Note that $\frac{STHT}{\Delta T} \leq 32767$	s	Real number	0.0	U
	+58 +59	GW	Gap width	0 to 100	%	Real number	0.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

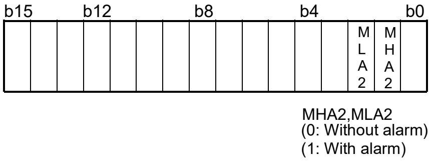
[*2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Specified position		Symbol	Name	Recommended range *1	Unit	Data format	Standard value	Set by
Loop tag memory *2	Ⓓ2+60 +61	GG	Gap gain	0 to 999999	—	Real number	1.0	U
	+62 +63	MVP	MV inside operation value	(-999999 to 999999)	%	Real number	0.0	S
Loop tag past value memory *2 *3	Ⓓ2+96 : +116	—	—	Used by the system as a work area.	—	—	—	—
Set value *4	Ⓢ3+0 +1	E2	Set value	-10 to 110	%	Real number	0.0	U

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

*2 The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

*3 The applications of the loop tag past value memory are indicated below.

Specified position	Description
Ⓓ2+96	Control cycle counter initial preset flag
+97	Sample counter*5
+98	Operation counter*5
+99	Hold counter
+100 +101	DV_{n-1} (Last deviation value)
+116	Alarm detection 2 (ALM2)  b15 b12 b8 b4 b0 MLA2 MHA2 MHA2,MLA2 (0: Without alarm) (1: With alarm)

When control is to be started from the initial status, the data must be cleared with the sequence program.

*4 The set value (E2) becomes valid when the set value pattern (SVPTN) is set to "E2 is used".
When using the MV of the upper loop as the set value (E2), specify the device where the manipulated value (MV) of the upper loop is set (offset + 12: MV).
When not using E2 as the set value, make sure to specify a dummy device.
(Special register SD1506 can be specified as a dummy device.)

*5 The sample counter and operation counter round off the data to the nearest whole number.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) SV setting processing

Either of the following processings is performed depending on the operation mode (MODE) setting.

(a) When the operation mode (MODE) is any of CAS, CCB and CSV

1) When the set value (E2) is specified, engineering value conversion is performed with the following expression and then "(2) Tracking processing" is performed.

$$SV_n = \frac{RH - RL}{100} \times E2 + RL$$

2) When the set value (E2) is not specified, "(2) Tracking processing" is performed without the engineering value conversion being performed.

(b) When the operation mode (MODE) is any of MAN, AUT, CMV, CMB, CAB, LCM, LCA and LCC, "(2) Tracking processing" is performed.

(2) Tracking processing

(a) The set value (SV) is converted reversely from the engineering value with the following operation expression to calculate SVn'.

$$SV_n' = \frac{100}{RH - RL} \times (SV_n - RL)$$

(b) When all of the following conditions hold, tracking processing is performed.

1) The tracking bit (TRK) of the operation constant is 1.

2) The set value (E2) is used.

3) The operation mode (MODE) is any of MAN, AUT, CMV, CMB, CAB, LCM, LCA and LCC.
E2=SVn'

(c) When the set value (E2) is the manipulated value (MV) of the upper loop, the tracking flag (TRKF) of the alarm detection inhibition (INH) in the upper loop turns to 1.

(3) Gain (Kp) operation processing

(a) The deviation (DV) is calculated under the following condition.

Condition	Operation expression
Forward operation(PN = 1)	DV = E1 - SVn'
Reverse operation(PN = 0)	DV = SVn' - E1

(b) The output gain (K) is calculated under the following condition.

Condition	Operation expression
When |DV| ≤ GW	K = GG
When |DV| > GW	$K = 1 - \frac{(1 - GG) \times GW}{\lvert DV \rvert}$

(4) SPI operation

SPI operation is performed with the following operation expression.

Condition	Operation expression
During operating time (ST)	$BW = K_P \times \{(DV_n - DV_{n-1}) + \frac{BT}{T_I} \times DV_n\}$
During hold time (STHT - ST)	BW = 0 (loop tag past value memory is not set.)

K_P: K × Gain (P), T_I: Integral constant (I), BT: Execution cycle (ΔT)

In the following case, however, note that special processing will be performed.

Condition			Processing
QnPHCPU/QnPRHCPU (First 5 digits of the serial No. : 07031 or earlier)	**QnPHCPU/QnPRHCPU (First 5 digits of the serial No. : 07032 or later)**	**QnUDPVCPU**	
In any of the following cases 1, 2, 3 1. Integral constant (I) = 0 (T_I = 0) 2. When either of MHA or MLA is turned to 1 (MVP > MH) and ($\frac{BT}{T_I} \times DV_n > 0$) 3. When either of MHA or MLA is turned to 1 (MVP < ML) and ($\frac{BT}{T_I} \times DV_n < 0$)	In any of the following cases 1, 2, 3 1. Integral constant (I) = 0 (T_I = 0) 2. When either of MHA2 or MLA2 is turned to 1 (MVP > MH) and ($\frac{BT}{T_I} \times DV_n > 0$) 3. When either of MHA2 or MLA 2 is turned to 1 (MVP < ML) and ($\frac{BT}{T_I} \times DV_n < 0$)		$\frac{BT}{T_I} \times DV_n = 0$

(5) Deviation check

A deviation check is made under the following condition and the result of the check is output to DVLA of the alarm detection (ALM) and the deviation large alarm (BB1) of the block memory.

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] When DVLI or ERRI of the alarm detection inhibition (INH) is 1, DVLA and BB1 turn to 0 since the alarm is inhibited.

(6) Loop stop processing

(a) Setting 1 in SPA of the alarm detection (ALM) selects a loop stop.

A loop stop performs the following processing and terminates the S.SPI instruction.

1) BW is turned to 0.

2) DVLA of the alarm detection (ALM) is turned to 0.

3) The operation mode (MODE) is changed to MAN.

4) BB1 of BB are turned to 0.

(b) Setting 0 in SPA of the alarm detection (ALM) selects a loop run.

A loop run performs "(7) Operating time/hold time check judgment ".

(7) Operating time/hold time check judgment

Whether it is the operating time (ST) or hold time (HT = STHT - ST) is judged and the following processing is performed.

(a) Operating time (ST)

SV setting processing, tracking processing, gain (Kp) operation processing, PI operation (operating time) and deviation check are performed.

(b) Hold time (HT = STHT - ST)

Tracking processing, SPI operation (hold time) and deviation check are performed.

Under the following condition, however, the hold time is set to 0 and continuous PI control is carried out.

$$\frac{STHT}{\Delta T} \leqq \frac{ST}{\Delta T}$$

When the integer part of $\frac{STHT}{\Delta T}$ is 0, no processing is performed. (ΔMV also remains unchanged.)

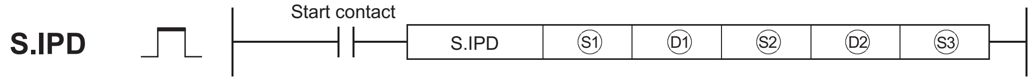
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

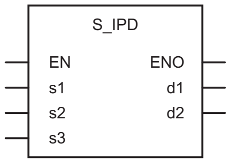
Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of (S1), (S2), (D2) are either a non-numeric or non-normalized number	○	○	—
	When the execution cycle (ΔT) < 0	○	○	○
	When the operating time (ST) < 0	○	○	○
	When the sample counter < 0	○	○	○
4140	When the values of (S1), (S2), (D2) are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

9.5 S.IPD

Ladder diagram



Structured ladder/FBD



Structured text language

```
ENO:=S_IPD(EN,s1,s2,s3,d1,d2);
```

Input argument	EN	: Execution condition	: Bit
	Ⓢ1	: Input data start device	: Real data type
	Ⓢ2	: Operation constant start device	: Array of any 16-bit data (0..6)
	Ⓢ3	: When set value (E2) is used: Set value start device When set value (E2) is not used: Dummy device[*1]	: Real data type
Output argument	ENO	: Execution result	: Bit
	Ⓓ1	: Block memory start device	: Array of any 16-bit data (0..2)
	Ⓓ2	: Loop tag memory start device	: Array of any 16-bit data (0..127)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
Ⓢ1	—	○		—					
Ⓓ1	—	○		—					
Ⓢ2	—	○		—					
Ⓓ2	—	○		—					
Ⓢ3	—	○		—					

[*1] Special register SD1506 can be specified as a dummy device.

Function

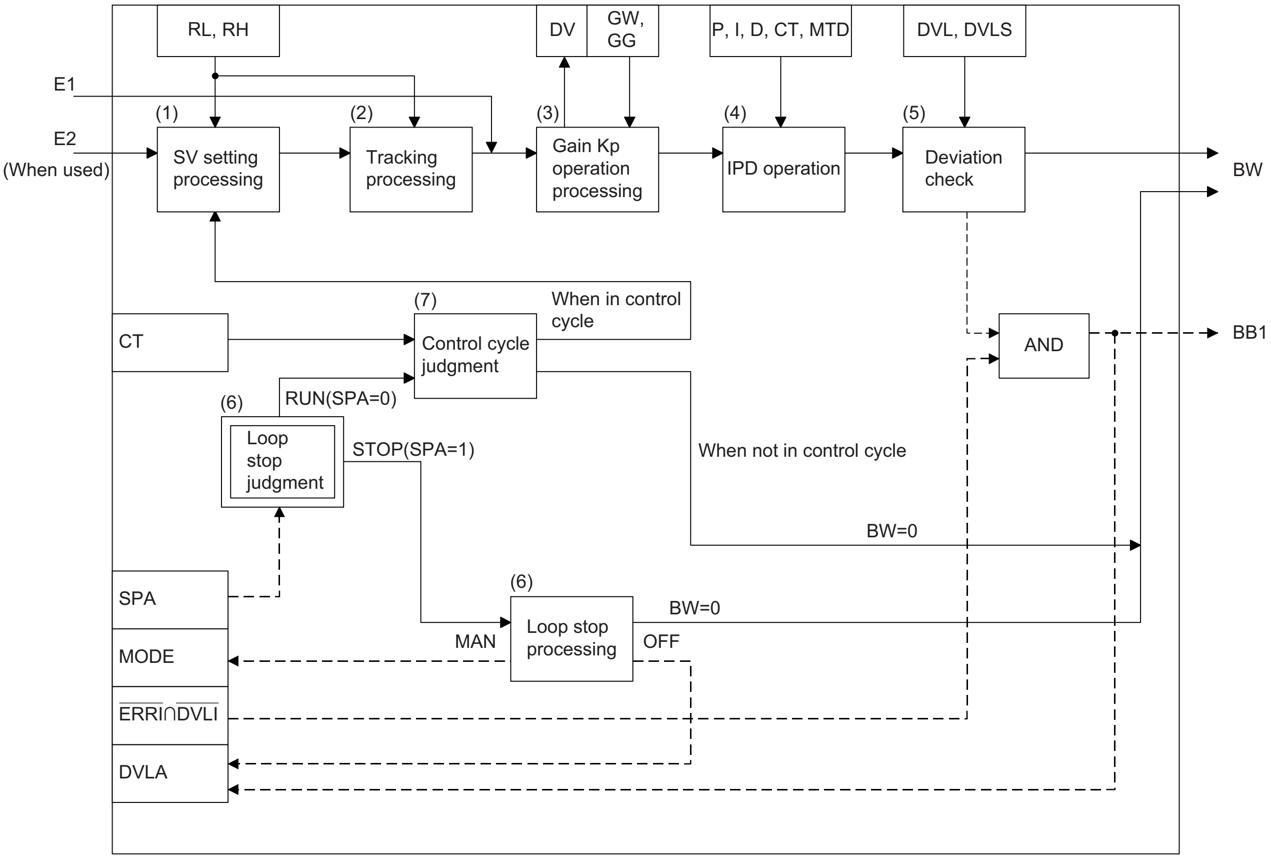
Performs I-PD control when the specified control cycle is reached.

Also performs SV setting processing, tracking processing, gain (Kp) operation processing and deviation check at this time.

Block diagram

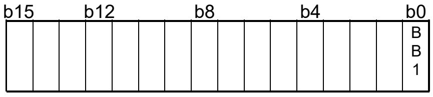
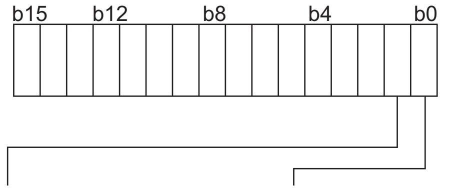
The processing block diagram of the S.IPD instruction is shown below.

(The numerals (1) to (7) in the diagram indicate the order of the processing.)



Set Data

(1) Data specified in S.IPD instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	S1+0 +1	E1	Input value	-999999 to 999999	%	Real number	—	U
Block memory	D1+0 +1	BW	Output value (Δ MV)	(-999999 to 999999)	%	Real number	—	S
	+2	BB	—					
		BB1	Deviation large alarm	 (0: Without alarm) (1: With alarm)	—	BIN 16bit	—	S
Operation constant	S2+0 +1	MTD	Derivative gain	0 to 999999	—	Real number	8.0	U
	+2 +3	DVLS	Deviation large alarm hysteresis	0 to 100	%	Real number	2.0	U
	+4	PN	Operation mode	0: Reverse operation 1: Forward operation	—	BIN 16bit	0	U
	+5	TRK	Tracking bit	0: Not trucked 1: Trucked	—	BIN 16bit	0	U
	+6	SVPTN	Set value pattern	0 to 3  Set value pattern[*3] 0: E2 is upper loop MV. 1: E2 is not upper loop MV. Set value used[*2] 0: E2 is used. 1: E2 is not used.	—	BIN 16bit	3	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[*2] Specify whether the set value (E2) is to be used or not.

[*3] Specify whether the MV of the upper loop is to be used or not as the set value (E2).

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Loop tag memory [*2]	Ⓓ2+1	MODE	Operation mode	0 to $FFFF_H$ b15 b12 b8 b4 b0 CSV CMV CCB CAB CMB CAS AUT MAN LCC LCA LCM	—	BIN 16bit	8_H	S/U
	+3	ALM	Alarm detection	0 to $FFFF_H$ b15 b12 b8 b4 b0 SPA DVLA MHA MLA SPA 0:Loop RUN 1:Loop STOP DVLA,MHA,MLA (0:Without alarm) (1:With alarm)	—	BIN 16bit	4000_H	S/U
	+4	INH	Alarm detection inhibition	00 to $FFFF_H$ b15 b12 b8 b4 b0 ERRI TRKF DVLI MHI MLI TRKF (0 : Without tracking) (1 : With tracking) ERRI, DVLI, MHI, MLI 0 : Alarm enable 1 : Alarm inhibit	—	BIN 16bit	0_H	S/U
	+14 +15	SV	Set value	RL to RH	—	Real number	0.0	U
	+16 +17	DV	Deviation	(-110 to 110)	%	Real number	0.0	S
	+22 +23	RH	Engineering value upper limit	-999999 to 999999	—	Real number	100.0	U
	+24 +25	RL	Engineering value lower limit	-999999 to 999999	—	Real number	0.0	U
	+46 +47	CT	Control cycle	0 to 999999 Note that $\frac{CT}{\Delta T} \leqq 32767$	s	Real number	1.0	U
	+50 +51	DVL	Deviation limit value	0 to 100	%	Real number	100.0	U
	+52 +53	P	Gain	0 to 999999	—	Real number	1.0	U
	+54 +55	I	Integral constant	0 to 999999	s	Real number	10.0	U
	+56 +57	D	Derivative constant	0 to 999999	s	Real number	0.0	U
	+58 +59	GW	Gap width	0 to 100	%	Real number	0.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

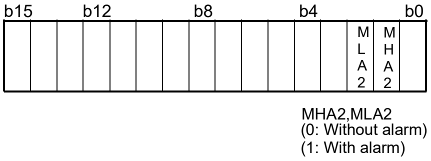
[*2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Specified position		Symbol	Name	Recommended range [1]	Unit	Data format	Standard value	Set by
Loop tag memory [2]	Ⓓ2+60 +61	GG	Gap gain	0 to 999999	—	Real number	1.0	U
	+62 +63	MVP	MV inside operation value	(-999999 to 999999)	%	Real number	0.0	S
Loop tag past value memory [2] [3]	Ⓓ2+96 ⋮ +116	—	—	Used by the system as a work area.	—	—	—	S
Set value [4]	Ⓢ3+0 +1	E2	Set value	-10 to 110	%	Real number	0.0	U

[1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

[3] The applications of the loop tag past value memory are indicated below.

Specified position	Description
Ⓓ2+96	Control cycle counter initial preset flag
+97	Control cycle counter[5]
+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} (Process value before last)
+116	Alarm detection 2 (ALM2)  b15 b12 b8 b4 b0 MLA2 MHA2 MHA2,MLA2 (0: Without alarm) (1: With alarm)

When control is to be started from the initial status, the data must be cleared with the sequence program.

[4] The set value (E2) becomes valid when the set value pattern (SVPTN) is set to "E2 is used".
When using the MV of the upper loop as the set value (E2), specify the device where the manipulated value (MV) of the upper loop is set (offset + 12: MV).
When not using E2 as the set value, make sure to specify a dummy device.
(Special register SD1506 can be specified as a dummy device.)

[5] The counrol cycle counter rounds off the data to the nearest whole number.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) SV setting processing

Either of the following processings is performed depending on the operation mode (MODE) setting.

(a) When the operation mode (MODE) is any of CAS, CCB and CSV

1) When the set value (E2) is specified, engineering value conversion is performed with the following expression and then "(2) Tracking processing" is performed.

$$SV_n = \frac{RH - RL}{100} \times E2 + RL$$

2) When the set value (E2) is not specified, "(2) Tracking processing" is performed without the engineering value conversion being performed.

(b) When the operation mode (MODE) is any of MAN, AUT, CMV, CMB, CAB, LCM, LCA and LCC, "(2) Tracking processing" is performed.

(2) Tracking processing

(a) The set value (SV) is converted reversely from the engineering value with the following operation expression to calculate SVn'.

$$SV_n' = \frac{100}{RH - RL} \times (SV_n - RL)$$

(b) When all of the following conditions hold, tracking processing is performed.

1) The tracking bit (TRK) of the operation constant is 1.

2) The set value (E2) is used.

3) The operation mode (MODE) is any of MAN, AUT, CMV, CMB, CAB, LCM, LCA and LCC.
E2 = SVn'

(c) When the set value (E2) is the manipulated value (MV) of the upper loop, the tracking flag (TRKF) of the alarm detection inhibition (INH) in the upper loop turns to 1.

(3) Gain (Kp) operation processing

(a) The deviation (DV) is calculated under the following condition.

Condition	Operation expression
Forward operation (PN = 1)	DV = E1 - SVn'
Reverse operation (PN = 0)	DV = SVn' - E1

(b) The output gain (K) is calculated under the following condition.

Condition	Operation expression
When $\lvert DV \rvert \leq GW$	K = GG
When $\lvert DV \rvert > GW$	$K = 1 - \frac{(1 - GG) \times GW}{\lvert DV \rvert}$

(4) I-PD operation

I-PD operation is performed with the following operation expression.

Item		Operation expression
Bn	When forward operation (PN = 1)	$B_{n-1}+\frac{M_D\times T_D}{M_D\times CT+T_D}\times\{(PV_n-2PV_{n-1}+PV_{n-2})-\frac{CT\times B_{n-1}}{T_D}\}$
	When reverse operation (PN = 0)	$B_{n-1}+\frac{M_D\times T_D}{M_D\times CT+T_D}\times\{-(PV_n-2PV_{n-1}+PV_{n-2})-\frac{CT\times B_{n-1}}{T_D}\}$
BW (ΔMV)	When forward operation (PN = 1)	$K_P\times\{\frac{CT}{T_I}\times DV_n+(PV_n-PV_{n-1})+B_n\}$
	When reverse operation (PN = 0)	$K_P\times\{\frac{CT}{T_I}\times DV_n-(PV_n-PV_{n-1})+B_n\}$

K_P: K × Gain (P), M_D: Derivative gain (MTD)

T_I: Integral constant (I), T_D: Derivative constant (D)

In the following case, however, note that special processing will be performed.

Condition			Processing
QnPHCPU/QnPRHCPU (First 5 digits of the serial No. : 07031 or	QnPHCPU/QnPRHCPU (First 5 digits of the serial No. : 07032 or later)	QnUDPVCPU	
In either of the following cases 1, 2 1. Derivative constant (D) = 0 (TD = 0) 2. Operation mode (MODE) is any of MAN, LCM and CMV			Bn = 0 (However, the loop tag past value memory is set.)
In any of the following cases 1, 2, 3 1. Integral constant (I) = 0 (T_I = 0) 2. When either of MHA or MLA error is turned to 1 (MVP > MH) and ($\frac{CT}{T_I}\times DV_n>0$) 3. When either of MHA or MLA error is turned to 1 (MVP < ML) and ($\frac{CT}{T_I}\times DV_n<0$)	In any of the following cases 1, 2, 3 1. Integral constant (I) = 0 (T_I = 0) 2. When either of MHA2 or MLA2 is turned to 1 (MVP > MH) and ($\frac{CT}{T_I}\times DV_n>0$) 3. When either of MHA2 or MLA 2 is turned to 1 (MVP < ML) and ($\frac{CT}{T_I}\times DV_n<0$)		$\frac{CT}{T_I}\times DV_n=0$

(5) Deviation check

A deviation check is made under the following condition and the result of the check is output to DVLA of the alarm detection (ALM) and the deviation large alarm (BB1) of the block memory.

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 When DVLI or ERRI in the alarm detection inhibition (INH) is set to 1, DVLA and BB1 show 0 since the alarm is prohibited.

(6) Loop stop processing

(a) Setting 1 in SPA of the alarm detection (ALM) selects a loop stop.

A loop stop performs the following processing and terminates the S.IPD instruction.

1) BW is turned to 0.

2) DVLA of the alarm detection (ALM) is turned to 0.

3) The operation mode (MODE) is changed to MAN.

4) BB1 of BB are turned to 0.

(b) Setting 0 in SPA of the alarm detection (ALM) selects a loop run.

A loop run performs "(7) Control cycle judgment".

(7) Control cycle judgment

(a) If the specified control cycle is not reached, BW (∆MV) is turned to 0 and the S.IPD instruction is terminated.

(b) When the specified control cycle is reached, "(1) SV setting processing" is performed.

Operation Error

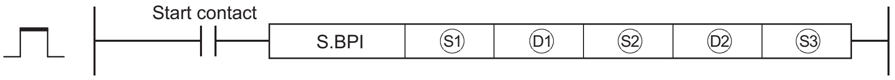
In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	○	○	—
	When CT < 0 or the execution cycle (SD1500) < 0	○	○	○
4140	When the values of Ⓢ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

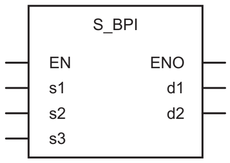
9.6 S.BPI

Ladder diagram

S.BPI



Structured ladder/FBD



Structured text language

ENO:=S_BPI(EN,s1,s2,s3,d1,d2);

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Real data type
	S2	: Operation constant start device	: Array of any 16-bit data (0..4)
	S3	: When set value (E2) is used: Set value start device When set value (E2) is not used: Dummy device*1	: Real data type
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Loop tag memory start device	: Array of any 16-bit data (0..127)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					
S3	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

Function

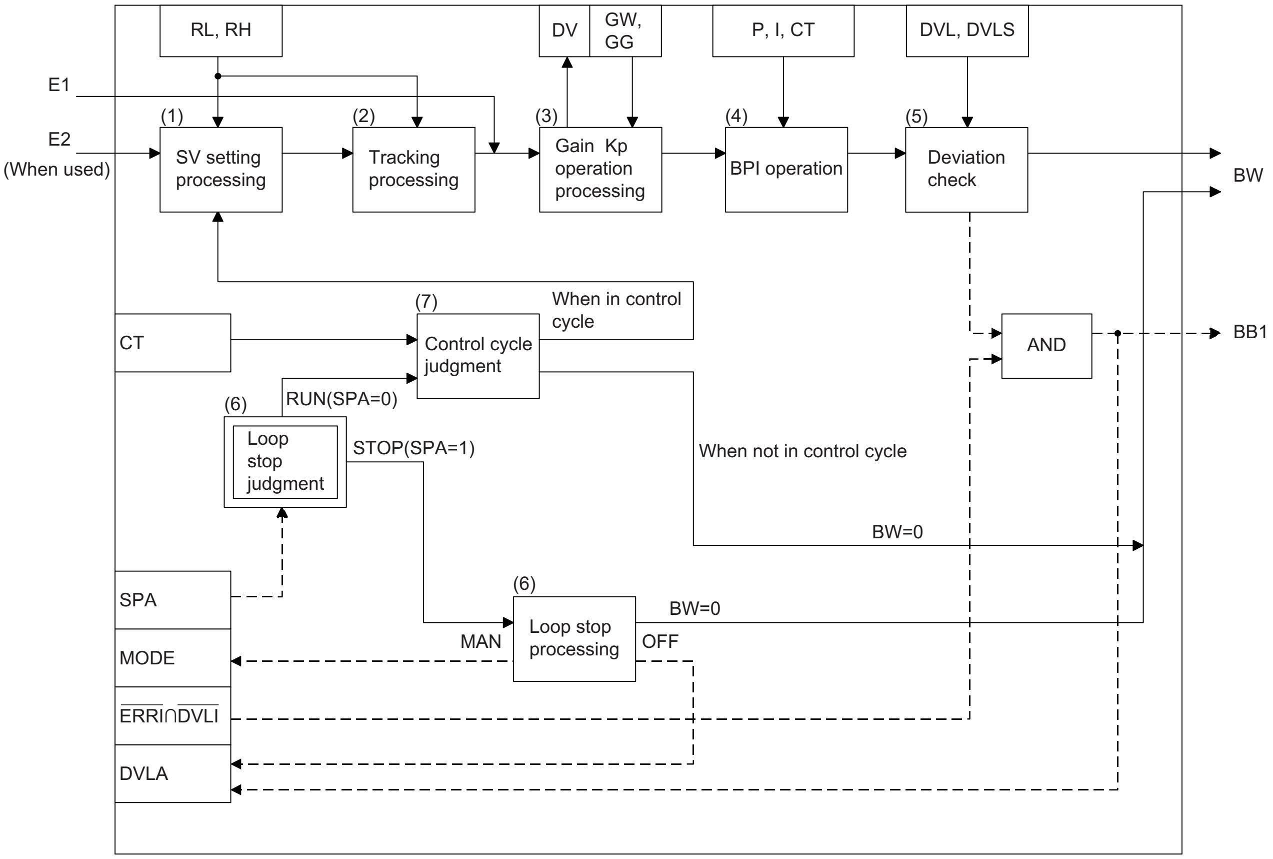
Performs BPI operation when the specified control cycle is reached.

Also performs SV setting processing, tracking processing, gain (Kp) operation processing and deviation check at this time.

Block diagram

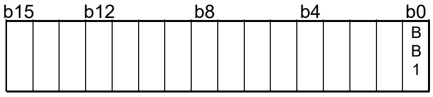
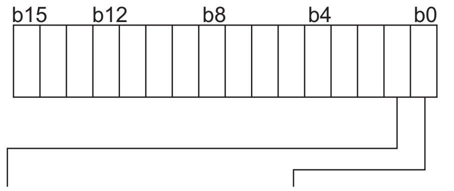
The processing block diagram of the S.BPI instruction is shown below.

(The numerals (1) to (7) in the diagram indicate the order of the processing.)



Set Data

(1) Data specified in S.BPI instruction

Specified position		Symbol	Name	Recommended range *1	Unit	Data format	Standard value	Set by
Input data	S1+0 +1	E1	Input value	-999999 to 999999	%	Real number	—	U
Block memory	D1+0 +1	BW	Output value (ΔMV)	(-999999 to 999999)	%	Real number	—	S
	+2	BB		—				
		BB1	Deviation large alarm	 (0: Without alarm) (1: With alarm)	—	BIN 16bit	—	S
Operation constant	S2+0 +1	DVLS	Deviation large alarm hysteresis	0 to 100	%	Real number	2.0	U
	+2	PN	Operation mode	0: Reverse operation 1: Forward operation	—	BIN 16bit	0	U
	+3	TRK	Tracking bit	0: Not trucked 1: Trucked	—	BIN 16bit	0	U
	+4	SNPTN	Set value pattern	0 to 3  Set value pattern*3 0: E2 is upper loop MV. 1: E2 is not upper loop MV. Set value used*2 0: E2 is used. 1: E2 is not used.	—	BIN 16bit	3	U

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

*2 Specify whether the set value (E2) is to be used or not.

*3 Specify whether the MV of the upper loop is to be used or not as the set value (E2).

<table>
<tr><th colspan="2">Specified position</th><th>Symbol</th><th>Name</th><th>Recommended range *1</th><th>Unit</th><th>Data format</th><th>Standard value</th><th>Set by</th></tr>
<tr><td rowspan="13">Loop tag memory *2</td><td>Ⓓ2+1</td><td>MODE</td><td>Operation mode</td><td>0 to FFFF$_H$
<table><tr><td>b15</td><td></td><td></td><td>b12</td><td></td><td></td><td></td><td>b8</td><td></td><td></td><td></td><td>b4</td><td></td><td></td><td></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>LCC</td><td>LCA</td><td>LCM</td></tr></table></td><td>—</td><td>BIN 16bit</td><td>8$_H$</td><td>S/U</td></tr>
<tr><td>+3</td><td>ALM</td><td>Alarm detection</td><td>0 to FFFF$_H$
<table><tr><td>b15</td><td></td><td></td><td>b12</td><td></td><td></td><td></td><td>b8</td><td></td><td></td><td></td><td>b4</td><td></td><td></td><td></td><td>b0</td></tr><tr><td></td><td>SPA</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>DVLA</td><td>MHA</td><td>MLA</td></tr></table>SPA
0:Loop RUN
1:Loop STOP
DVLA,MHA,MLA
(0:Without alarm)
(1:With alarm)</td><td>—</td><td>BIN 16bit</td><td>4000$_H$</td><td>S/U</td></tr>
<tr><td>+4</td><td>INH</td><td>Alarm detection inhibition</td><td>00 to FFFF$_H$
<table><tr><td>b15</td><td></td><td></td><td>b12</td><td></td><td></td><td></td><td>b8</td><td></td><td></td><td></td><td>b4</td><td></td><td></td><td></td><td>b0</td></tr><tr><td>ERRI</td><td></td><td>TRKF</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>DVLI</td><td>MHI</td><td>MLI</td></tr></table>TRKF
(0 : Without tracking)
(1 : With tracking)
ERRI, DVLI, MHI, MLI
0 : Alarm enable
1 : Alarm inhibit</td><td>—</td><td>BIN 16bit</td><td>0$_H$</td><td>S/U</td></tr>
<tr><td>+14
+15</td><td>SV</td><td>Set value</td><td>RL to RH</td><td>—</td><td>Real number</td><td>0.0</td><td>U</td></tr>
<tr><td>+16
+17</td><td>DV</td><td>Deviation</td><td>(-110 to 110)</td><td>%</td><td>Real number</td><td>0.0</td><td>S</td></tr>
<tr><td>+22
+23</td><td>RH</td><td>Engineering value upper limit</td><td>-999999 to 999999</td><td>—</td><td>Real number</td><td>100.0</td><td>U</td></tr>
<tr><td>+24
+25</td><td>RL</td><td>Engineering value lower limit</td><td>-999999 to 999999</td><td>—</td><td>Real number</td><td>0.0</td><td>U</td></tr>
<tr><td>+46
+47</td><td>CT</td><td>Control cycle</td><td>0 to 999999 Note that $\frac{CT}{\Delta T} \leqq 32767$</td><td>s</td><td>Real number</td><td>1.0</td><td>U</td></tr>
<tr><td>+50
+51</td><td>DVL</td><td>Deviation limit value</td><td>0 to 100</td><td>%</td><td>Real number</td><td>100.0</td><td>U</td></tr>
<tr><td>+52
+53</td><td>P</td><td>Gain</td><td>0 to 999999</td><td>—</td><td>Real number</td><td>1.0</td><td>U</td></tr>
<tr><td>+54
+55</td><td>I</td><td>Integral constant</td><td>0 to 999999</td><td>s</td><td>Real number</td><td>10.0</td><td>U</td></tr>
<tr><td>+56
+57</td><td>SDV</td><td>DV cumulative value (Σ DV)</td><td>(-999999 to 999999)</td><td>%</td><td>Real number</td><td>0.0</td><td>S</td></tr>
</table>

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

*2 The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Specified position		Symbol	Name	Recommended range *1	Unit	Data format	Standard value	Set by
Loop tag Memory *2	Ⓓ2+58 +59	GW	Gap width	0 to 100	%	Real number	0.0	U
	+60 +61	GG	Gap gain	0 to 999999	—	Real number	1.0	U
Loop tag past value Memory *2 *3	Ⓓ2+96 ⋮ +99	—	—	Used by the system as a work area.	—	—	—	S
Set value *4	Ⓢ3+0 +1	E2	Set value	-10 to 110	%	Real number	0.0	U

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

*2 The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

*3 The applications of the loop tag past value memory are indicated below.

Specified position	Description
Ⓓ2+96	Control cycle initial preset flag
+97	Control cycle counter*5
+98 +99	$\frac{CT}{T_I} \times \Sigma DV_I$

When control is to be started from the initial status, the data must be cleared with the sequence program.

*4 The set value (E2) becomes valid when the set value pattern (SVPTN) is set to "E2 is used".
When using the MV of the upper loop as the set value (E2), specify the device where the manipulated value (MV) of the upper loop is set (offset + 12: MV).
When not using E2 as the set value, make sure to specify a dummy device.
(Special register SD1506 can be specified as a dummy device.)

*5 The control cycle counter rounds off the data to the nearest whole number.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) SV setting processing

Either of the following processings is performed depending on the operation mode (MODE) setting.

(a) When the operation mode (MODE) is any of CAS, CCB and CSV

1) When the set value (E2) is specified, engineering value conversion is performed with the following expression and then "(2) Tracking processing" is performed.

$$SV_n = \frac{RH - RL}{100} \times E2 + RL$$

2) When the set value (E2) is not specified, "(2) Tracking processing" is performed without the engineering value conversion being performed.

(b) When the operation mode (MODE) is any of MAN, AUT, CMV, CMB, CAB, LCM, LCA and LCC, "(2) Tracking processing" is performed.

(2) Tracking processing

(a) The set value (SV) is converted reversely from the engineering value with the following operation expression to calculate SVn'.

$$SV_n' = \frac{100}{RH - RL} \times (SV_n - RL)$$

(b) When all of the following conditions hold, tracking processing is performed.

1) The tracking bit (TRK) of the operation constant is 1.

2) The set value (E2) is used.

3) The operation mode (MODE) is any of MAN, AUT, CMV, CMB, CAB, LCM, LCA and LCC.
E2 = SVn'

(c) When the set value (E2) is the manipulated value (MV) of the upper loop, the tracking flag (TRKF) of the alarm detection inhibition (INH) in the upper loop turns to 1.

(3) Gain (Kp) operation processing

(a) The deviation (DV) is calculated under the following condition.

Condition	Operation expression
Forward operation (PN = 1)	DV = E1 - SVn'
Reverse operation (PN = 0)	DV = SVn' - E1

(b) The output gain (K) is calculated under the following condition.

Condition	Operation expression
When |DV| ≤ GW	K = GG
When |DV| > GW	$K = 1 - \frac{(1 - GG) \times GW}{\lvert DV \rvert}$

(4) BPI operation

BPI operation is performed with the following operation expression.

Condition	Operation expression
BW (ΔMV)	$K_P \times BW \times (DV_n + \frac{CT}{T_I} \times \Sigma DV_I)$

Kp: K × Gain (P), BT: Execution cycle, TI: Integral constant (I),
Σ DVI: Cumulative value of DVn, DVn: Deviation

In the following case, however, note that special processing will be performed.

Condition	Processing
In either of the following cases 1, 2 1. Integral constant (I) = 0 (TI = 0) 2. Either MLA or MHA of alarm detection (ALM) is 1	$\frac{CT}{T_I} \times \Sigma DV_I =$ last value unchanged
1. Integral constant (I) ≠ 0 (TI ≠ 0)	$\frac{CT}{T_I} \times \Sigma DV_I = \frac{CT}{T_I} \times (\Sigma DV_I + DV_n)$

(5) Deviation check

A deviation check is made under the following condition and the result of the check is output to DVLA of the alarm detection (ALM) and the deviation large alarm (BB1) of the block memory.

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] When DVLI or ERRI in the alarm detection inhibition (INH) is set to 1, DVLA and BB1 show 0 since the alarm is prohibited.

(6) Loop stop processing

(a) Setting 1 in SPA of the alarm detection (ALM) selects a loop stop.
A loop stop performs the following processing and terminates the S.BPI instruction.

1) BW is turned to 0.
2) DVLA of the alarm detection (ALM) is turned to 0.
3) The operation mode (MODE) is changed to MAN.
4) BB1 of BB are turned to 0.

(b) Setting 0 in SPA of the alarm detection (ALM) selects a loop run.
A loop run performs "(7) Control cycle judgment".

(7) Control cycle judgment

(a) If the specified control cycle is not reached, BW is turned to 0 and the S.BPI instruction is terminated.

(b) When the specified control cycle is reached, "(1) SV setting processing" is performed.

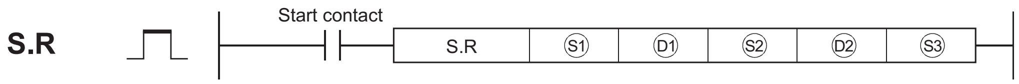
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

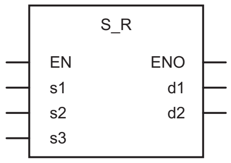
Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	○	○	—
	When CT < 0 or the execution cycle (SD1500) < 0	○	○	○
4140	When the values of Ⓢ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

9.7 S.R

Ladder diagram



Structured ladder/FBD



Structured text language

ENO:=S_R(EN,s1,s2,s3,d1,d2);

Input argument	EN	: Execution condition	: Bit
	Ⓢ1	: Input data start device	: Real data type
	Ⓢ2	: Operation constant start device	: Array of any 16-bit data (0..1)
	Ⓢ3	: When set value (E2) is used: Set value start device When set value (E2) is not used: Dummy device*1	: Real data type
Output argument	ENO	: Execution result	: Bit
	Ⓓ1	: Block memory start device	: Real data type
	Ⓓ2	: Loop tag memory start device	: Array of any 16-bit data (0..127)

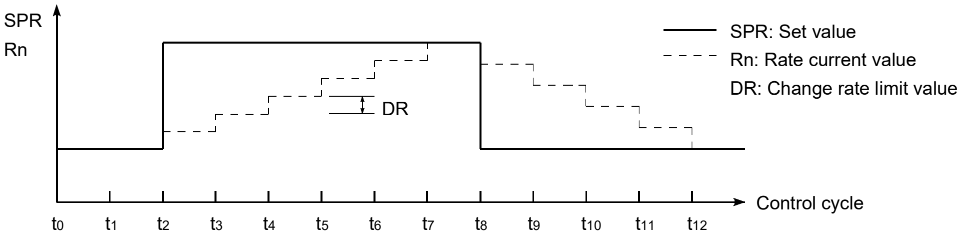
Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
Ⓢ1	—	○		—					
Ⓓ1	—	○		—					
Ⓢ2	—	○		—					
Ⓓ2	—	○		—					
Ⓢ3	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

Function

Performs rate operation when the specified control cycle is reached.

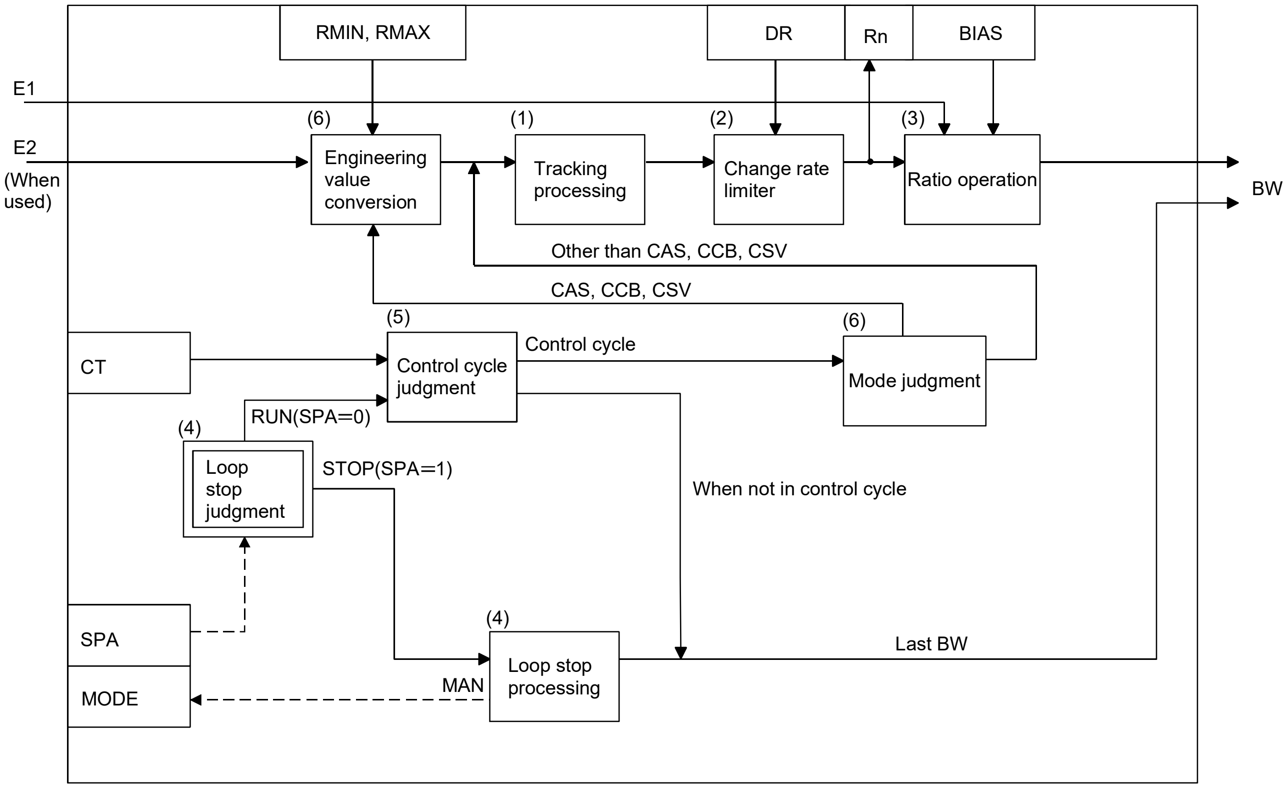
Also performs operation mode (MODE) judgment, engineering value conversion, tracking processing and change rate limiter processing at this time.



Block diagram

The processing block diagram of the S.R instruction is shown below.

(The numerals (1) to (6) in the diagram indicate the order of the processing.)



Set Data

(1) Data specified in S.R instruction

Specified position		Symbol	Name	Recommended range [1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value	-999999 to 999999	%	Real number	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	%	Real number	—	S
Operation constant	Ⓢ2+0	TRK	Tracking bit	0: Not trucked 1: Trucked	—	BIN 16bit	0	U
	+1	SVPTN	Set value pattern	0 to 3 b15 b12 b8 b4 b0 Set value pattern[3] 0: E2 is upper loop MV. 1: E2 is not upper loop MV. Set value used[2] 0: E2 is used. 1: E2 is not used.	—	BIN 16bit	3	U
Loop tag memory [4]	Ⓓ2+1	MODE	Operation mode	0 to FFFF$_H$ b15 b12 b8 b4 b0 CSV CMV CCB CAB CMB CAS AUT MAN LCC LCA LCM	—	BIN 16bit	8$_H$	S/U
	+3	ALM	Alarm detection	0 to FFFF$_H$ b15 b12 b8 b4 b0 SPA SPA 0: Loop RUN 1: Loop STOP	—	BIN 16bit	4000$_H$	S/U
	+14 +15	SPR	Set value	-999999 to 999999	—	Real number	0.0	U
	+16 +17	BIAS	Bias	-999999 to 999999	%	Real number	0.0	U
	+46 +47	CT	Control cycle	0 to 999999 Note that $\frac{CT}{\Delta T} \leqq 32767$	s	Real number	1.0	U
	+50 +51	DR	Change rate limit value	0 to 999999	—	Real number	100.0	U

[1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[2] Specify whether the set value (E2) is to be used or not.

[3] Specify whether the MV of the upper loop is to be used or not as the set value (E2).

[4] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Specified position		Symbol	Name	Recommended range *1	Unit	Data format	Standard value	Set by
Loop tag memory *2	Ⓓ2+52 +53	RMAX	Rate upper limit value	-999999 to 999999	—	Real number	100.0	U
	+54 +55	RMIN	Rate lower limit value	-999999 to 999999	—	Real number	0.0	U
	+56 +57	R_n	Rate current value	(-999999 to 999999)	—	Real number	0.0	S
Loop tag past value memory *2 *3	Ⓓ2+96 ⋮ +99	—	—	Used by the system as a work area.	—	—	—	S
Set value *4	Ⓢ3+0 +1	E2	Set value	-10 to 110	%	Real number	0.0	U

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

*2 The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

*3 The applications of the loop tag past value memory are indicated below.

Specified position	Description
Ⓓ2+96	Control cycle counter initial preset flag
+97	Control cycle counter*5
+98 +99	R_{n-1} (Last value)

When control is to be started from the initial status, the data must be cleared with the sequence program.

*4 The set value (E2) becomes valid when the set value pattern (SVPTN) is set to "E2 is used".
When using the MV of the upper loop as the set value (E2), specify the device where the manipulated value (MV) of the upper loop is set (offset + 12: MV).
When not using E2 as the set value, make sure to specify a dummy device.
(Special register SD1506 can be specified as a dummy device.)

*5 The control cycle counter rounds off the data to the nearest whole number.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) Tracking processing

(a) When all of the following conditions hold, tracking processing is performed.

1) The tracking bit (TRK) of the operation constant is 1.

2) The set value (E2) is used.

3) The operation mode (MODE) is any of MAN, AUT, CMV, CMB, CAB, LCM, LCA and LCC.

$$E2 = \frac{100}{RMAX\text{-}RMIN} \times (SPR\text{-}RMIN)$$

(b) When the set value (E2) is the manipulated value (MV) of the upper loop, the tracking flag (TRKF) of the alarm detection inhibition (INH) in the upper loop turns to 1.

(2) Change rate limiter

In the change rate limiter, the following operation is performed and the result of the operation is stored into the current rate value (Rn).

Condition	Operation expression
$(SPR - R_n) \geq DR$	$R_n = R_{n-1} + DR$
$(SPR - R_n) \leq - DR$	$R_n = R_{n-1} - DR$
$\|SPR - R_n\| < DR$	$R_n = SPR$

(3) Rate operation

Rate operation is performed with the following operation expression.

$$BW = \frac{R_n - RMIN}{RMAX - RMIN} \times E1 + BIAS$$

(4) Loop stop processing

(a) Setting 1 in SPA of the alarm detection (ALM) selects a loop stop.

A loop stop performs the following processing and terminates the S.R instruction.

1) BW retains the last value.

2) The operation mode (MODE) is changed to MAN.

(b) Setting 0 in SPA of the alarm detection (ALM) selects a loop run.

A loop run performs "(5) Control cycle judgment".

(5) Control cycle judgment

(a) When the specified control cycle is not reached, BW is retained and the S.R instruction is terminated.

(b) When the specified control cycle is reached, "(6) Mode judgment" is performed.

(6) Mode judgment

Either of the following processings is performed depending on the operation mode (MODE) setting.

(a) When the operation mode (MODE) is any of CAS, CCB and CSV

1) When the set value (E2) is specified, engineering value conversion is made with the following expression and then "(2) Change rate limiter" is performed.

$$SPR = \frac{RMAX - RMIN}{100} \times E2 + RMIN$$

2) When the set value (E2) is not specified, "(2) Change rate limiter" is performed without engineering value conversion being made.

(b) When the operation mode (MODE) is any of MAN, AUT, CMV, CMB, CAB, LCM, LCA and LCC, "(1) Tracking processing" is executed.

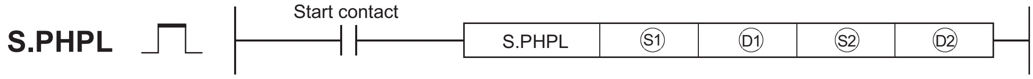
Operation Error

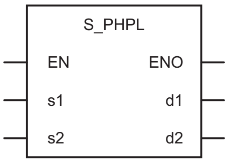
In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓓ2 are either a non-numeric or non-normalized number	○	○	—
	When CT < 0 or the execution cycle (SD1500) < 0	○	○	○
4140	When the values of Ⓢ1, Ⓓ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

9.8 S.PHPL

Ladder diagram



Structured ladder/FBD	Structured text language
	ENO:=S_PHPL(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	Ⓢ1	: Input data start device	: Real data type
	Ⓢ2	: Dummy device*1	: Real data type
Output argument	ENO	: Execution result	: Bit
	Ⓓ1	: Block memory start device	: Array of any 16-bit data (0..2)
	Ⓓ2	: Loop tag memory start device	: Array of any 16-bit data (0..127)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
Ⓢ1	—	○		—					
Ⓓ1	—	○		—					
Ⓢ2	—	○		—					
Ⓓ2	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

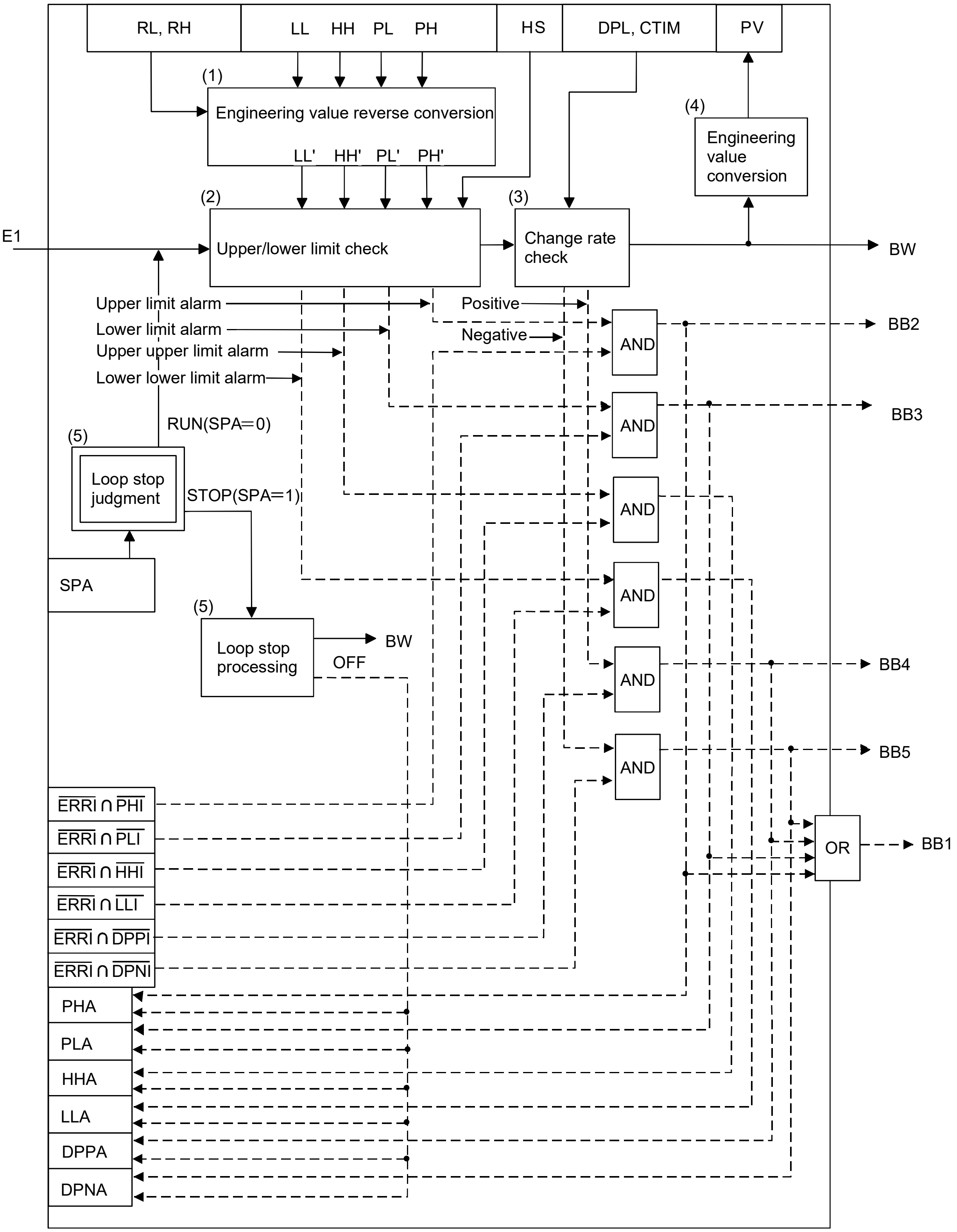
Function

Performs a high/low limit check on the input value (E1) and provides an alarm output.

Block diagram

The processing block diagram of the S.PHPL instruction is shown below.

(The numerals (1) to (5) in the diagram indicate the order of the processing.)



Set Data

(1) Data specified in S.PHPL instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	S1+0 +1	E1	Input value	-999999 to 999999	%	Real number	—	U
Block memory	D1+0 +1	BW	Output value	(-999999 to 999999)	%	Real number	—	S
	+2	BB	—					
		BB1	Alarm	b15 b12 b8 b4 b0 b4: BB5, b3: BB4, b2: BB3, b1: BB2, b0: BB1 (0: Without alarm) (1: With alarm)	—	BIN 16bit	—	S
		BB2	Upper limit alarm					
		BB3	Lower limit alarm					
		BB4	Positive direction change rate alarm					
		BB5	Negative direction change rate alarm					
Loop tag memory [*2]	D2+1	MODE	Operation mode	0 to $FFFF_H$ b15 b12 b8 b4 b0 b10: CSV, b9: CMV, b8: CCB, b7: CAB, b6: CMB, b5: CAS, b4: AUT, b3: MAN, b2: LCC, b1: LCA, b0: LCM	—	BIN 16bit	8_H	S/U
	+3	ALM	Alarm detection	0 to $FFFF_H$ b15 b12 b8 b4 b0 b14: SPA, b8: HHA, b7: LLA, b6: PHA, b5: PLA, b4: DPPA, b3: DPNA SPA 0: Loop RUN 1: Loop STOP Other (0: Without alarm) (1: With alarm)	—	BIN 16bit	4000_H	S/U
	+4	INH	Alarm detection inhibition	0 to $FFFF_H$ b15 b12 b8 b4 b0 b15: ERRI, b8: HHI, b7: LLI, b6: PHI, b5: PLI, b4: DPPI, b3: DPNI 0: Alarm enable 1: Alarm inhibit	—	BIN 16bit	0_H	S/U
	+10 +11	PV	Process value	(RL to RH)	—	Real number	0.0	S
	+22 +23	RH	Engineering value upper limit	-999999 to 999999	—	Real number	100.0	U
	+24 +25	RL	Engineering value lower limit	-999999 to 999999	—	Real number	0.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[*2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Specified position		Symbol	Name	Recommended range *1	Unit	Data format	Standard value	Set by
Loop tag memory *2	Ⓓ2+26 +27	PH	Upper limit alarm set value	RL to RH	—	Real number	100.0	U
	+28 +29	PL	Lower limit alarm value	RL to RH	—	Real number	0.0	U
	+30 +31	HH	Upper upper limit alarm value	RL to RH	—	Real number	100.0	U
	+32 +33	LL	Lower lower limit alarm value	RL to RH	—	Real number	0.0	U
	+40 +41	HS	Upper/lower limit alarm hysteresis	0 to 999999	%	Real number	0.0	U
	+42 +43	CTIM	Change rate alarm Check time	0 to 999999 Note that $\frac{CTIM}{\Delta T} \leqq 32767$	s	Real number	0.0	U
	+44 +45	DPL	Change rate alarm value	0 to 100	%	Real number	100.0	U
Loop tag past value memory *2 *3	Ⓓ2+124 ⋮ +127	—	—	Used by the system as a work area.	—	—	—	S

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

*2 The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

*3 The applications of the loop tag past value memory are indicated below.

Specified position	Description
Ⓓ2+124	Change rate monitor counter initial preset flag
+125	Change rate monitor counter*4
+126 +127	$E1_{n-m}$

When control is to be started from the initial status, the data must be cleared with the sequence program.

*4 The change rate monitor counter rounds off the data to the nearest whole number.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) Engineering value reverse conversion

The following operations are performed to match the upper limit alarm value (PH), lower limit alarm value (PL), upper upper limit alarm value (HH) and lower lower limit alarm value (LL) ranges with the input value (E1).

$$PH' = \frac{100}{RH - RL} \times (PH - RL), \qquad PL' = \frac{100}{RH - RL} \times (PL - RL)$$

$$HH' = \frac{100}{RH - RL} \times (HH - RL), \qquad LL' = \frac{100}{RH - RL} \times (LL - RL)$$

(2) Upper/lower limit check

The upper/lower limit checks of the input value (E1) are made under the following conditions.

Check item	Condition	ALM	BB2	BB3
Upper limit check	E1 > PH'	PHA = 1 [*1]	1 [*1]	—
	E1 ≤ PH' - HS	PHA = 0	0	—
	Others	PHA: Last value is status hold [*1]	Hold [*1]	—
Lower limit check	E1 < PL'	PLA = 1 [*2]	—	1 [*2]
	E1 ≥ PL' + HS	PLA = 0	—	0
	Others	PLA: Last value is status hold [*2]	—	Hold [*2]
Upper Upper limit check	E1 > HH'	HHA = 1 [*3]	—	—
	E1 ≤ HH' - HS	HHA = 0	—	—
	Others	HHA: Last value is status hold [*3]	—	—
Lower lower limit check	E1 < LL'	LLA = 1 [*4]	—	—
	E1 ≥ LL' + HS	LLA = 0	—	—
	Others	LLA: Last value is status hold [*4]	—	—

[*1] When PHI or ERRI in the alarm detection inhibition (INH) is set to 1, PHA and BB2 show 0 since the alarm is prohibited.

[*2] When PLI or ERRI in the alarm detection inhibition (INH) is set to 1, PLA and BB3 show 0 since the alarm is prohibited.

[*3] When HHI or ERRI in the alarm detection inhibition (INH) is set to 1, HHA show 0 since the alarm is prohibited.

[*4] When LLI or ERRI in the alarm detection inhibition (INH) is set to 1, LLA show 0 since the alarm is prohibited.

(3) Change rate check

(a) A change rate check is performed for the time specified in CTIM.

The number of change rate checks to be made is found by the following expression.

$$m = \frac{CTIM}{\Delta T}$$

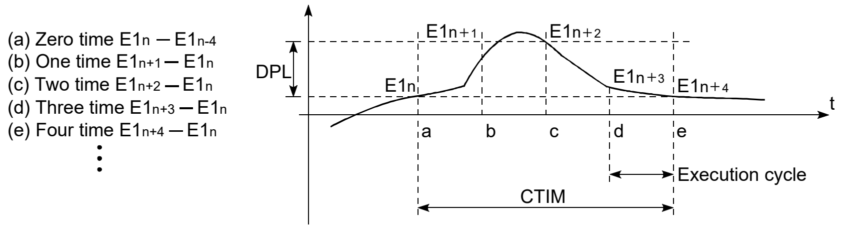
m varies from 1 to m.

However, when m = 0 (integer part), no processing is performed.

Set values of CTIM and ΔT so that the change rate alarm counter (m) ≥ 2.

When m = 0 (integer part), no processing is performed.

For example, when m = 4, the processing is performed as shown below.


(a) Zero time $E1_{n} - E1_{n-4}$
(b) One time $E1_{n+1} - E1_{n}$
(c) Two time $E1_{n+2} - E1_{n}$
(d) Three time $E1_{n+3} - E1_{n}$
(e) Four time $E1_{n+4} - E1_{n}$
⋮
DPL
$E1_{n}$
$E1_{n+1}$
$E1_{n+2}$
$E1_{n+3}$
$E1_{n+4}$
t
a
b
c
d
e
Execution cycle
CTIM

(b) The change of the input data is compared with the change rate alarm value (DPL) in each execution cycle (ΔT).

Check item	Condition	ALM	BB4	BB5
Change 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 [*2]	—	1 [*2]
	Others	DPNA = 0	—	0

[*1] When DPPI or ERRI in the alarm detection inhibition (INH) is set to 1, DPPA and BB4 show 0 since the alarm is prohibited.

[*2] When DPNI or ERRI in the alarm detection inhibition (INH) is set to 1, DPNA and BB5 show 0 since the alarm is prohibited.

(4) Engineering value conversion

Engineering value conversion is made with the following expression.

$$PV = \frac{RH - RL}{100} \times E1 + RL$$

(5) Loop stop processing

(a) Setting 1 in SPA of the alarm detection (ALM) selects a loop stop.

A loop stop performs the following processing and terminates the S.PHPL instruction.

1) Engineering value reverse conversion is performed with the following expression.

$$BW = \frac{100}{RH - RL} \times (PV - RL)$$

2) BB1 to BB5 of BB are turned to 0.

3) DPNA, DPPA, LLA, HHA, PLA and PHA of the alarm detection (ALM) are turned to 0.

(b) Setting 0 in SPA of the alarm detection (ALM) selects a loop run.

A loop run performs "(1) Engineering value reverse conversion".

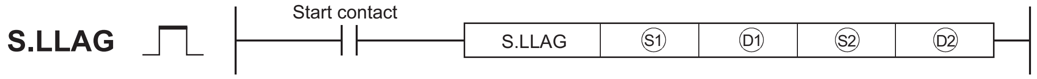
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

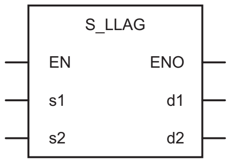
Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓓ2 are either a non-numeric or non-normalized number	○	○	—
	When the execution cycle (SD1500) < 0	○	○	○
	When DPL < -DPL	○	○	○
	When CTIM < 0	○	○	○
4140	When the values of Ⓢ1, Ⓓ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

9.9 S.LLAG

Ladder diagram



Structured ladder/FBD



Structured text language

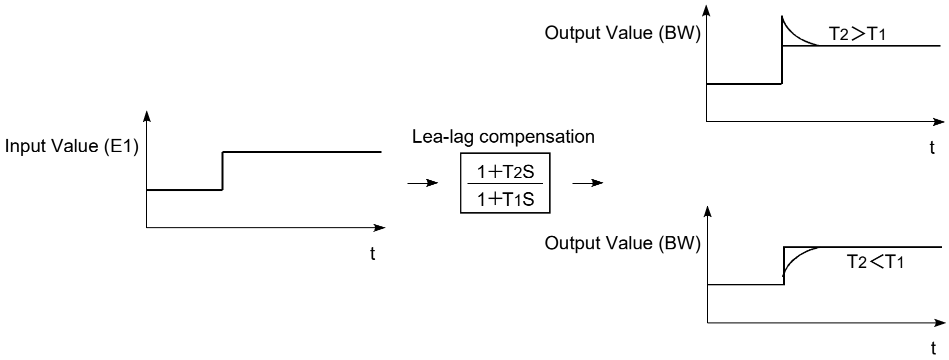
ENO:=S_LLAG(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	Ⓢ1	: Input data start device	: Array of any 16-bit data (0..2)
	Ⓢ2	: Operation constant start device	: Array of real data type (0..1)
Output argument	ENO	: Execution result	: Bit
	Ⓓ1	: Block memory start device	: Real data type
	Ⓓ2	: Loop tag memory start device	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
Ⓢ1	—	○		—					
Ⓓ1	—	○		—					
Ⓢ2	—	○		—					
Ⓓ2	—	○		—					

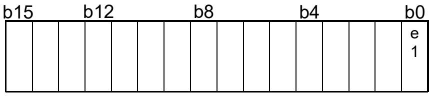
Function

Performs lead-lag operation according to the lag time and lead time settings of the operation constants and the actuating signal (e1).



Set Data

(1) Data specified in S.LLAG instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value	-999999 to 999999	%	Real number	—	U
	+2	e1	Actuating signal	 0: With lead-lag compensation 1: Without lead-lag compensation	—	BIN 16bit	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	%	Real number	—	S
Operation constant	Ⓢ2+0 +1	T_1	Delay time	0 to 999999	s	Real number	1.0	U
	+2 +3	T_2	Lead time	0 to 999999	s	Real number	1.0	U
Local work memory [*2]	Ⓓ2+0 +1	$E1_{n-1}$	Last Input value	Used by the system as a work area.	—	Real number	—	S

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[*2] When control is to be started from the initial status, the data must be cleared with the sequence program.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

The S.LLAG instruction instructs the following operation.

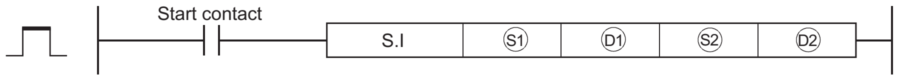
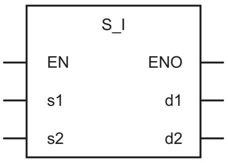
Condition	BW (Output value)
e1 = 0	$BW = \frac{1}{T_1+\Delta T} \times \{T_2 \times (E1 - E1_{n-1}) + T_1 \times (\text{BW Last value}) + \Delta T \times E1\}$ However, when $T_1 + \Delta T = 0$, BW = 0
e1 = 1	BW = E1 (Input value is output unchanged)

Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓓ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	○	○	—
	When the execution cycle (SD1500, SD1501) < 0	○	○	○
	When the lag time (T_1) < 0 or the lead time (T_2) < 0	○	○	○
4140	When the values of Ⓢ1, Ⓓ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

9.10 S.I

<table>
<tr><th colspan="2">Ladder diagram</th></tr>
<tr><td colspan="2">S.I 
</td></tr>
<tr><th>Structured ladder/FBD</th><th>Structured text language</th></tr>
<tr><td>
</td><td>ENO:=S_I(EN,s1,s2,d1,d2);</td></tr>
</table>

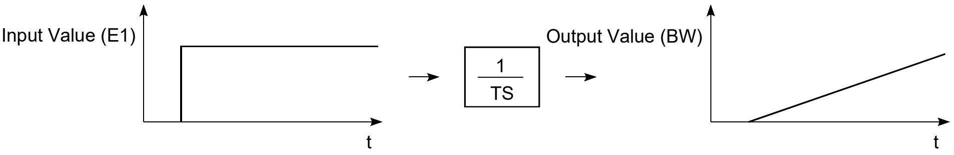
Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Array of any 16-bit data (0..2)
	S2	: Operation constant start device	: Array of real data type (0..1)
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Real data type
	D2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

[*1] Special register SD1506 can be specified as a dummy device.

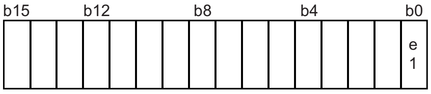
Function

Performs integral operation according to the operation control signal (e1).



Set Data

(1) Data specified in S.I instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value	-999999 to 999999	—	Real number	—	U
	+2	e1	Operation control signal	 0: With integral operation 1: Without integral operation	—	BIN 16bit	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	—	Real number	—	S
Operation constant	Ⓢ2+0 +1	T	Integral time	0 to 999999	s	Real number	1.0	U
	+2 +3	Ys	Output initial value	-999999 to 999999	—	Real number	0.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

The S.I instruction performs the following operation.

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: Current input value, ΔT: Execution cycle, Yn: Current output value, Y_{n-1}: Last output value

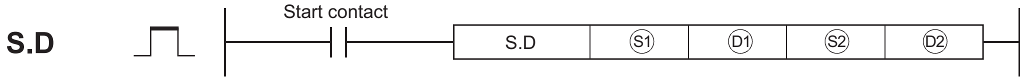
Operation Error

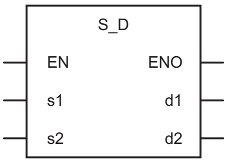
In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓓ1, Ⓢ2 are either a non-numeric or non-normalized number	○	○	—
4140	When the values of Ⓢ1, Ⓓ1, Ⓢ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

9.11 S.D

Ladder diagram



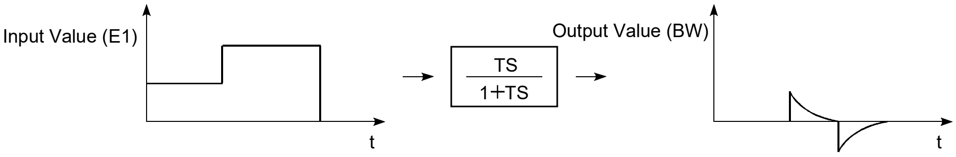
Structured ladder/FBD	Structured text language
	ENO:=S_D(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Array of any 16-bit data (0..2)
	S2	: Operation constant start device	: Array of real data type (0..1)
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Real data type
	D2	: Loop work memory start device	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

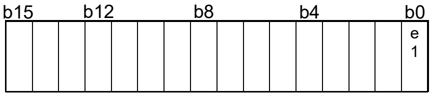
Function

Executes derivative operation according to the operation control signal (e1).



Set Data

(1) Data specified in S.D instruction

Specified position		Symbol	Name	Recommended range *1	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value	-999999 to 999999	—	Real number	—	U
	+2	e1	Operation control signal	 0: With derivative operation 1: Without derivative operation	—	BIN 16bit	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	—	Real number	—	S
Operation constant	Ⓢ2+0 +1	T	Derivative time	0 to 999999	s	Real number	1.0	U
	+2 +3	Ys	Output initial value	-999999 to 999999	—	Real number	0.0	U
Local work memory *2	Ⓓ2+0 +1	$E1_{n-1}$	Last input value	Used by the system as a work area.	—	Real number	—	S

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

*2 When control is to be started from the initial status, the data must be cleared with the sequence program.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

The S.D instruction performs the following operation.

e1	BW
0	$BW = \frac{T}{T+\Delta T} \times (Y_{n-1}-E1_{n-1}+E1)$ Note that T + ΔT = 0, BW = 0.
1	$BW = Y_s$

E1: Current input value, ΔT: Execution cycle, Y_n: Last output value, Y_{n-1}: Last output value

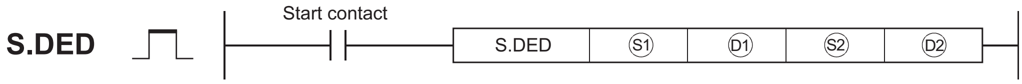
Operation Error

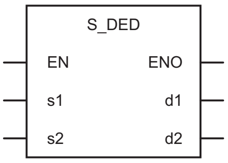
In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓓ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	○	○	—
4140	When the values of Ⓢ1, Ⓓ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

9.12 S.DED

Ladder diagram



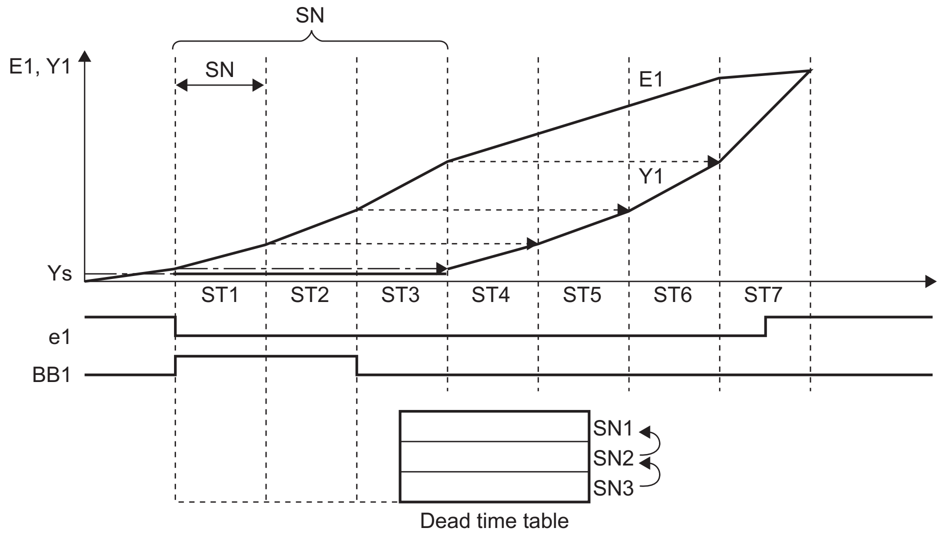
Structured ladder/FBD	Structured text language
	ENO:=S_DED(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Array of any 16-bit data (0..2)
	S2	: Operation constant start device	: Array of any 16-bit data (0..5)
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Loop work memory start device	: Array of any 16-bit data (0..100)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

Function

Outputs the input value (E1) with a delay of dead time according to the setting of the operation control signal (e1).

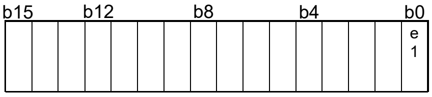
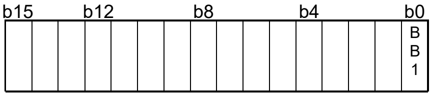
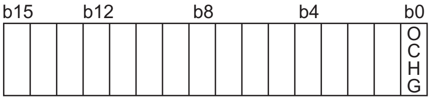


SN: Sampling count E1: Input value

ST: Data collection interval Ys: Output initial value

Set Data

(1) Data specified in S.DED instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value	-999999 to 999999	—	Real number	—	U
	+2	e1	Operation control signal	 0: With dead time 1: Without dead time	—	BIN 16bit	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	—	Real number	—	S
	+2	BB	—					
		BB1	Data sufficiency bit	 (0: Data sufficiency) (1: Data insufficiency)	—	BIN 16bit	—	S
Operation constant	Ⓢ2+0 +1	ST	Data collection Interval	0 to 999999 Note that $\frac{ST}{\Delta T} \leqq 32767$	s	Real number	1.0	U
	+2	SN	Sampling count	0 to 48	—	BIN 16bit	0	U
	+3 +4	Ys	Output initial value	-999999 to 999999	—	Real number	0.0	U
	+5	OCHG	output switching	 0: E1 when e1 turned from 1 to 0 is output up to SN times. 1: Ys is output up to SN times.	—	BIN 16bit	0	U
Local work memory [*2]	Ⓓ2+0	—	Last value input (e1')	Used by the system as a work area.	—	—	—	S
	+1		Cycle counter [*3]					
	+2		Dead time table number of stored data					
	+3 +4		Dead time table 1					
	+5 +6		Dead time table 2					
	:		:					
	+2SN +1 +2SN +2		Dead time table SN					

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[*2] When control is to be started from the initial status, the data must be cleared with the sequence program.

[*3] The cycle counter rounds off the data to the nearest whole number.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) The S.DED instruction performs the following operation.

e1	OCHG	Dead time	BW	
1	0/1	None	E1	
1→0	0	ST × SN	Up to SN times	E1 when e1 turns from 1 to 0
			Later than SN times	Oldest data [*1]
	1		Up to SN times	Ys
			Later than SN times	Oldest data [*1]
0→0	0/1	ST × SN	Oldest data [*1]	

*1 The oldest date is the E1 after the SNth time.

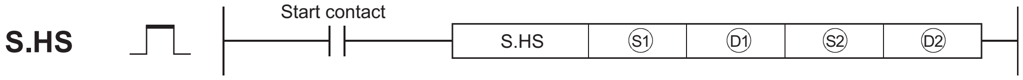
- When the dead time table date is not filled, BB1 is turned 1.
- When SN = 0, BB1 = 0 and BW = E1.

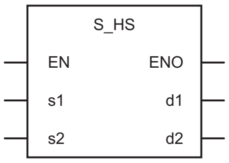
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	○	○	—
	When the sampling count is other than 0 to 48	○	○	○
	When the execution cycle (ΔT) < 0	○	○	○
	When the data collection intervals (ST) < 0	○	○	○
4140	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

9.13 S.HS

Ladder diagram


Structured ladder/FBD	Structured text language
	ENO:=S_HS(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	Ⓢ1	: Input data start device	: Array of any 16-bit data (0..32)
	Ⓢ2	: Dummy device[*1]	: Real data type
Output argument	ENO	: Execution result	: Bit
	Ⓓ1	: Block memory start device	: Array of any 16-bit data (0..2)
	Ⓓ2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
Ⓢ1	—	○		—					
Ⓓ1	—	○		—					
Ⓢ2	—	○		—					
Ⓓ2	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

Function

Outputs the maximum value of the input values 1 (E1) to n (En).

Set Data

(1) Data specified in S.HS instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	S1+0	n	Input count	1 to 16	—	BIN 16bit	—	U
	+1 +2	E1	Input value 1	-999999 to 999999	—	Real number	—	U
	+3 +4	E2	Input value 2					
	⋮	⋮	⋮					
	+2n-1 +2n	En	Input value n					
Block memory	D1+0 +1	BW	Output value	(Maximum value of E1 to En)	—	Real number	—	S
	+2	BB	—					
		BB1 to BB16	Output selection	b15 b12 b8 b4 b0 BB16 BB15 BB14 BB13 BB12 BB11 BB10 BB9 BB8 BB7 BB6 BB5 BB4 BB3 BB2 BB1 (0: The corresponding input value is not the maximum value) (1: The corresponding input value is the maximum value)	—	BIN 16bit	—	S

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) High selector processing

The maximum value of the input values 1 (E1) to n (En) is stored into BW.

Also, any of BB1 to BB16 of BB corresponding to the maximum value is turned to 1.

Input value	E16	E15	E14	to	E2	E1
Bit turned to 1 at maximum value	BB16	BB15	BB14	to	BB2	BB1

(a) If there are two or more maximum values, the bits corresponding to the maximum values are all turned to 1.

(b) If there is only one input

1) When only E1 is used as the input value
   - E1 is stored into BW.
   - BB1 of BB is turned to 1.
   - BB2 to BB16 of BB are turned to 0.

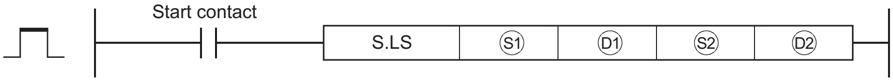
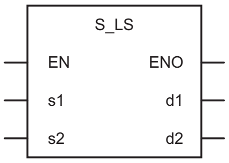
2) Only one of E2 to E16 is used as the input value
   - The input values of E2 to E16 and the data of E1 are used to perform processing.

Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When the value of S1 is either a non-numeric or non-normalized number	○	○	—
	When not 1 ≤ number of inputs (n) ≤ 16	○	○	○
4140	When the value of S1 is either a non-numeric or non-normalized number	—	—	○

9.14 S.LS

Ladder diagram	
S.LS	
Structured ladder/FBD	**Structured text language**
	ENO:=S_LS(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Array of any 16-bit data (0..32)
	S2	: Dummy device[*1]	: Real data type
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

Function

Outputs the minimum value of the input values 1 (E1) to n (En).

Set Data

(1) Data specified in S.LS instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0	n	Input count	1 to 16	—	BIN 16bit	—	U
	+1 +2	E1	Input value 1	-999999 to 999999	—	Real number	—	U
	+3 +4	E2	Input value 2					
	⋮	⋮	⋮					
	+2n-1 +2n	En	Input value n					
Block memory	Ⓓ1+0 +1	BW	Output value	(Minimum value of E1 to En)	—	Real number	—	S
	+2	BB	—					
		BB1 to BB16	Output selection	b15 b12 b8 b4 b0 BB16 BB15 BB14 BB13 BB12 BB11 BB10 BB9 BB8 BB7 BB6 BB5 BB4 BB3 BB2 BB1 (0: The corresponding input value is not the minimum value) (1: The corresponding input value is the minimum value)	—	BIN 16bit	—	S

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) Low selector processing

The minimum value of the input values 1 (E1) to n (En) is stored into BW.

Also, any of BB1 to BB16 of BB corresponding to the minimum value is turned to 1.

Input value	E16	E15	E14	to	E2	E1
Bit turned to 1 at minimum value	BB16	BB15	BB14	to	BB2	BB1

(a) If there are two or more minimum values, the bits corresponding to the minimum values are all turned to 1.

(b) If there is only one input

1) When only E1 is used as the input value

- E1 is stored into BW.
- BB1 of BB is turned to 1.
- BB2 to BB16 of BB are turned to 0.

2) Only one of E2 to E16 is used as the input value

- The input values of E2 to E16 and the data of E1 are used to perform processing.

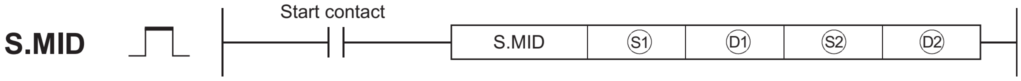
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

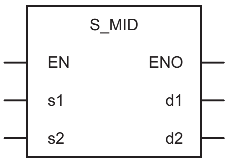
Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When the value of Ⓢ1 is either a non-numeric or non-normalized number	○	○	—
	When not 1 ≤ number of inputs (n) ≤ 16	○	○	○
4140	When the value of Ⓢ1 is either a non-numeric or non-normalized number	—	—	○

9.15 S.MID

Ladder diagram



Structured ladder/FBD



Structured text language

ENO:=S_MID(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Array of any 16-bit data (0..32)
	S2	: Dummy device[*1]	: Real data type
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

Function

Outputs the middle value between the maximum value and minimum value among the input value 1 (E1) to input value n (En).

Set Data

(1) Data specified in S.MID instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0	n	Input count	1 to 16	—	BIN 16bit	—	U
	+1 +2	E1	Input value 1	-999999 to 999999	—	Real number	—	U
	+3 +4	E2	Input value 2					
	⋮	⋮	⋮					
	+2n-1 +2n	En	Input value n					
Block memory	Ⓓ1+0 +1	BW	Output value	(Middle value between maximum value and minimum value)	—	Real number	—	S
	+2	BB	—					
		BB1 to BB16	Output selection	b15 b12 b8 b4 b0 BB16 BB15 BB14 BB13 BB12 BB11 BB10 BB9 BB8 BB7 BB6 BB5 BB4 BB3 BB2 BB1 (0: The corresponding input value is not the intermediate value) (1: The corresponding input value is the intermediate value)	—	BIN 16bit	—	S

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) Middle value selector processing

The middle value of the input values 1 (E1) to n (En) is stored into BW.

Also, any of BB1 to BB16 of BB corresponding to the middle value is turned to 1.

Input value	E16	E15	E14	to	E2	E1
Bit turned to 1 at middle value	BB16	BB15	BB14	to	BB2	BB1

(a) If there are an even number of inputs, the smaller value of the middle values is stored.

(b) If there are two or more middle values, the bits corresponding to the middle values are all turned to 1.

Remark

The middle value is selected as described below.

(1) The input value 1 (E1) to input value n (En) are rearranged in order of increasing value.
(If there are the same input values, they are arranged in order of increasing input number.)

(2) The middle value among the rearranged values is selected.

Example) When the input data are 2, 5, 1, 4 and 3, the middle value is selected as described below.

Input data

E1	E2	E3	E4	E5
2	5	1	4	3

Rearrangement →

Rearranged data

E3	E1	E5	E4	E2
1	2	3	4	5

In the above case, the middle value is "3" and BB5 turns to 1.

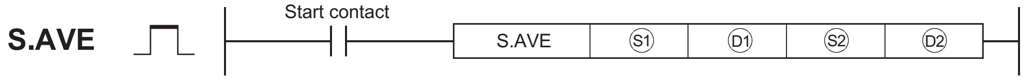
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

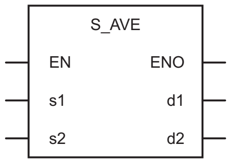
Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When the value of Ⓢ1 is either a non-numeric or non-normalized number	○	○	—
	When not 1 ≤ number of inputs (n) ≤ 16	○	○	○
4140	When the value of Ⓢ1 is either a non-numeric or non-normalized number	—	—	○

9.16 S.AVE

Ladder diagram



Structured ladder/FBD



Structured text language

ENO:=S_AVE(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	Ⓢ1	: Input data start device	: Array of any 16-bit data (0..32)
	Ⓢ2	: Dummy device[*1]	: Real data type
Output argument	ENO	: Execution result	: Bit
	Ⓓ1	: Block memory start device	: Array of any 16-bit data (0..2)
	Ⓓ2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
Ⓢ1	—	○		—					
Ⓓ1	—	○		—					
Ⓢ2	—	○		—					
Ⓓ2	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

Function

Calculates and outputs the average value of the input value 1 (E1) to n (En).

Set Data

(1) Data specified in S.AVE instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0	n	Input count	1 to 16	—	BIN 16bit	—	U
	+1 +2	E1	Input value 1	-999999 to 999999	—	Real number	—	U
	+3 +4	E2	Input value 2					
	⋮	⋮	⋮					
	+2n-1 +2n	E_n	Input value n					
Block memory	Ⓓ1+0 +1	BW	Output value	(Average value of E1 to En)	—	Real number	—	S

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) Calculation of average value

The average value of the input value 1 (E1) to n (En) is calculated.

As the denominator (N), the value specified as the number of inputs (n) is used.

$$BW = \frac{E1 + E2 + E3 + \cdots\cdot E_n}{N}$$

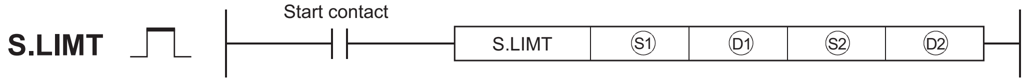
Operation Error

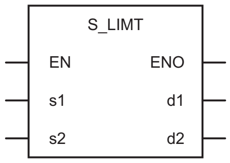
In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the value of Ⓢ1 is either a non-numeric or non-normalized number	○	○	—
	When not 1 ≤ number of inputs (n) ≤ 16	○	○	○
4140	When the value of Ⓢ1 is either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

9.17 S.LIMT

Ladder diagram



Structured ladder/FBD	Structured text language
	ENO:=S_LIMT(EN,s1,s2,d1,d2);

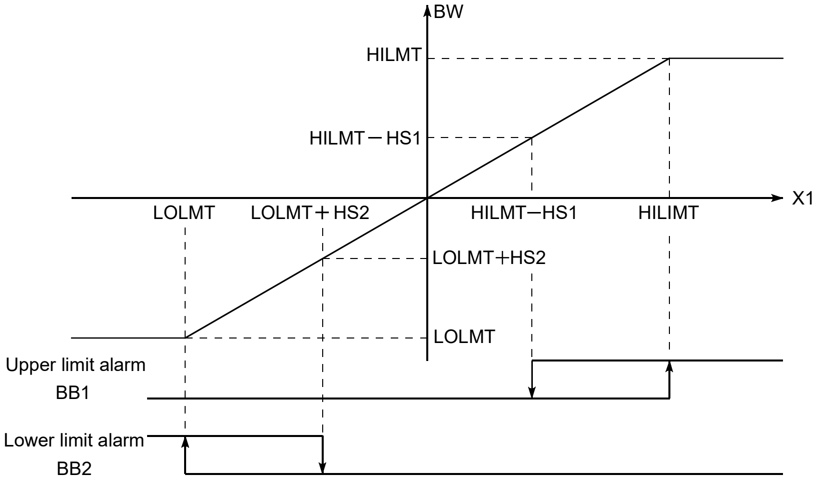
Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Real data type
	S2	: Operation constant start device	: Array of real data type (0..3)
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

Function

The upper and lower limit limiter is applied to the output value by adding a hysteresis.



Set Data

(1) Data specified in S.LIMT instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value	-999999 to 999999	%	Real number	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	%	Real number	—	S
	+2	BB	—					
		BB1	Upper limit alarm	b15 b12 b8 b4 b0 BB2 BB1 (0: Without alarm) (1: With alarm)	—	BIN 16bit	—	S
		BB2	Lower limit alarm					
Operation constant	Ⓢ2+0 +1	HILMT	Upper limit value[*2]	-999999 to 999999	%	Real number	100.0	U
	+2 +3	LOLMT	Lower limit value[*2]	-999999 to 999999	%	Real number	0.0	U
	+4 +5	HS1	Upper limit hysteresis	0 to 999999	%	Real number	0.0	U
	+6 +7	HS2	Lower limit hysteresis	0 to 999999	%	Real number	0.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[*2] Make setting to satisfy HILMT ≥ LOLMT.

Processing contents

(1) The S.LIMT instruction performs the following operation.

Condition	BW	BB1	BB2
E1 ≥ HILMT	HILMT	1	0
(LOLMT + HS2) < E1 < (HILMT - HS1)	E1	0	0
E1 ≤ LOLMT	LOLMT	0	1
Other than above (hysteresis section)	E1	Last value	Last value

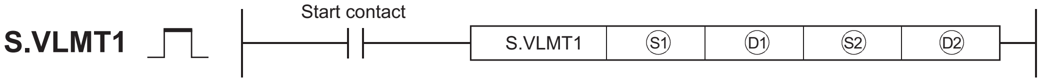
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

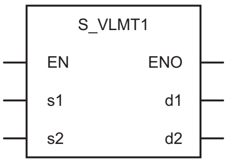
Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When upper limit hysteresis (HS1) or lower limit hysteresis (HS2) is a negative value	○	○	—
	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	○	○	—
	When LOLMT > HILMT	○	○	○
4140	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○
	When upper limit hysteresis (HS1) or lower limit hysteresis (HS2) is a negative value	—	—	○

9.18 S.VLMT1

Ladder diagram



Structured ladder/FBD



Structured text language

ENO:=S_VLMT1(EN,s1,s2,d1,d2);

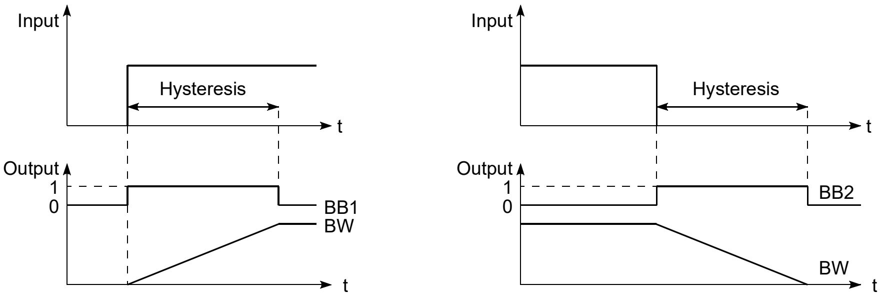
Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Real data type
	S2	: Operation constant start device	: Array of real data type (0..3)
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

Function

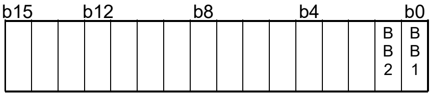
Limits the varying speed of the output value.



(Positive Direction) (Negative Direction)

Set Data

(1) Data specified in S.VLMT1 instruction

Specified position		Symbol		Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1		Input value	-999999 to 999999	%	Real number	—	U
Block memory	Ⓓ1+0 +1	BW		Output value	(-999999 to 999999)	%	Real number	—	S
	+2	BB		—					
			BB1	Positive direction restriction alarm	 (0: Without alarm) (1: With alarm)	—	BIN 16bit	—	S
			BB2	Negative direction restriction alarm					
Operation constant	Ⓢ2+0 +1	V1		Positive direction limit value	0 to 999999	%/s	Real number	100.0	U
	+2 +3	V2		Negative direction limit value	0 to 999999	%/s	Real number	100.0	U
	+4 +5	HS1		Positive direction hysteresis	0 to 999999	%	Real number	0.0	U
	+6 +7	HS2		Negative direction hysteresis	0 to 999999	%	Real number	0.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) The S.VLMT1 instruction performs the following operation.

	Input (E1 - BW)	BW	BB1	BB2
Positive direction When E1 ≥ 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 When 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

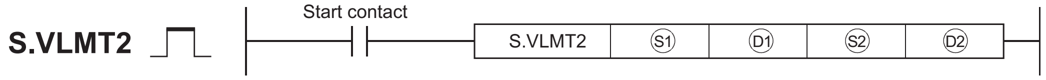
Operation Error

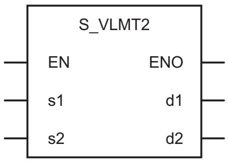
In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	○	○	—
	When HS1 < 0 or HS2 < 0	○	○	○
4140	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

9.19 S.VLMT2

Ladder diagram



Structured ladder/FBD	Structured text language
	ENO:=S_VLMT2(EN,s1,s2,d1,d2);

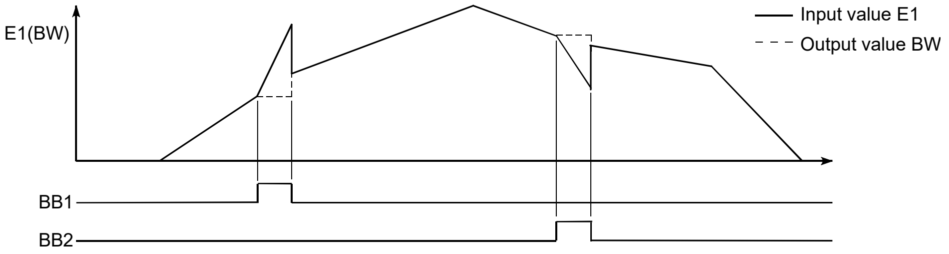
Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Real data type
	S2	: Operation constant start device	: Array of real data type (0..3)
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

[*1] Special register SD1506 can be specified as a dummy device.

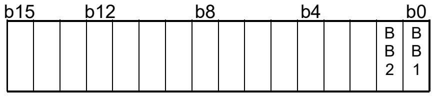
Function

Limits the varying speed of the output value.



Set Data

(1) Data specified in S.VLMT2 instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	S1+0 +1	E1	Input value	-999999 to 999999	%	Real number	—	U
Block memory	D1+0 +1	BW	Output value	(-999999 to 999999)	%	Real number	—	S
	+2	BB	—					
		BB1	Positive direction restriction alarm	 (0: Without alarm) (1: With alarm)	—	BIN 16Bit	—	S
		BB2	Negative direction restriction alarm					
Operation constant	S2+0 +1	V1	Positive direction limit value	0 to 999999	%/s	Real number	100.0	U
	+2 +3	V2	Negative direction limit value	0 to 999999	%/s	Real number	100.0	U
	+4 +5	HS1	Positive direction hysteresis	0 to 999999	%	Real number	0.0	U
	+6 +7	HS2	Negative direction hysteresis	0 to 999999	%	Real number	0.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) The S.VLMT2 instruction performs the following operation.

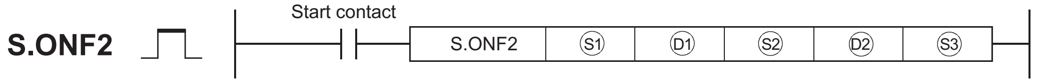
	Condition	BW	BB1	BB2
Positive direction When E1 ≥ BW	(E1 - BW) ≥ (V1× ΔT)	BW = BW	1	0
	(E1 - BW) < (V1× ΔT - HS1)	BW = E1	0	0
	Others	BW = BW	Last value	Last value
Negative direction When E1 < BW	(BW - E1) ≥ (V2× ΔT)	BW = BW	0	1
	(BW - E1) < (V2× ΔT - HS2)	BW = E1	0	0
	Others	BW = BW	Last value	Last value

Operation Error

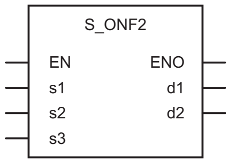
In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	○	○	—
	When HS1 < 0 or HS2 < 0	○	○	○
4140	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

9.20 S.ONF2

Ladder diagram


S.ONF2
Start contact
S.ONF2
S1
D1
S2
D2
S3

Structured ladder/FBD	Structured text language
	ENO:=S_ONF2(EN,s1,s2,s3,d1,d2);

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Real data type
	S2	: Operation constant start device	: Array of any 16-bit data (0..2)
	S3	: When set value (E2) is used: Set value start device When set value (E2) is not used: Dummy device[*1]	: Real data type
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Loop tag memory start device	: Array of any 16-bit data (0..127)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					
S3	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

Function

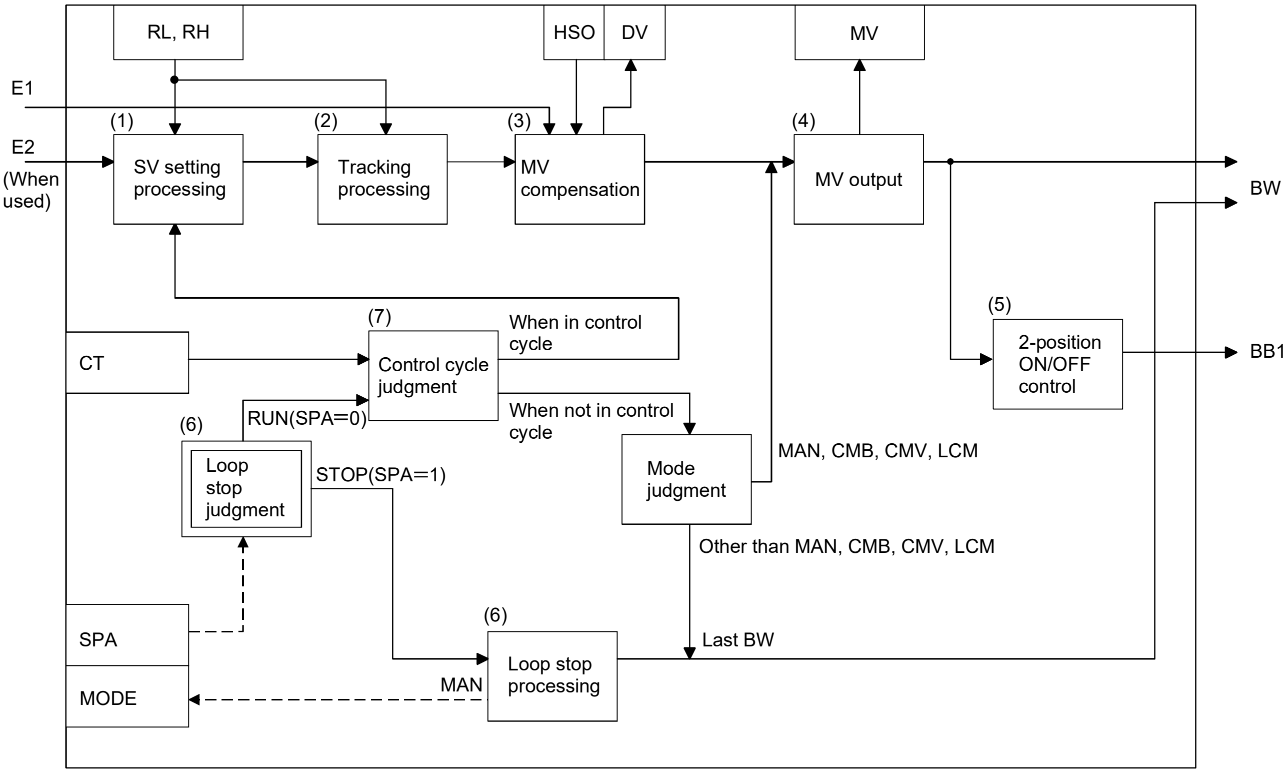
Performs 2-position ON/OFF control (ON/OFF of one contact) when the specified control cycle is reached.

Also performs SV setting processing, tracking processing, MV compensation and MV output processing at this time.

Block diagram

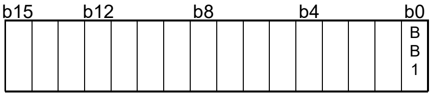
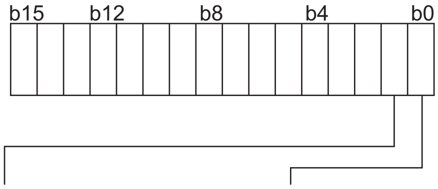
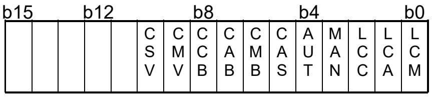
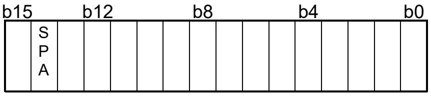
The processing block diagram of the S.ONF2 instruction is shown below.

(The numerals (1) to (7) in the diagram indicate the order of the processing.)



Set Data

(1) Data specified in S.ONF2 instruction

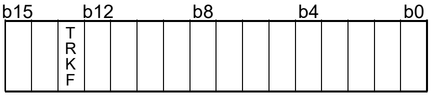
Specified position		Symbol	Name	Recommended range *1	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value	-999999 to 999999	%	Real number	—	S/U
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	%	Real number	—	S
	+2	BB	—					
		BB1	Operation result	 (0: |BW|＜50%) (1: |BW|≧50%)	—	BIN 16bit	—	S
Operation constant	Ⓢ2+0	PN	Operation mode	0: Reverse operation 1: Forward operation	—	BIN 16bit	0	U
	+1	TRK	Tracking bit	0: Without tracking 1: With tracking	—	BIN 16bit	0	U
	+2	SVPTN	Set value pattern	0 to 3  Set value pattern*3 0: E2 is upper loop MV. 1: E2 is not upper loop MV. Set value used*2 0: E2 is used. 1: E2 is not used.	—	BIN 16bit	3	U
Loop tag memory *4	Ⓓ2+1	MODE	Operation mode	0 to FFFF$_H$  b15 b12 b8 b4 b0 CSV CMV CCB CAB CMB CAS AUT MAN LCC LCA LCM	—	BIN 16bit	8$_H$	S/U
	+3	ALM	Alarm detection	0 to FFFF$_H$  SPA 0: Loop RUN 1: Loop STOP	—	BIN 16bit	4000$_H$	S/U

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

*2 Specify whether the set value (E2) is to be used or not.

*3 Specify whether the MV of the upper loop is to be used or not as the set value (E2).

*4 The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Loop tag memory [*2]	Ⓓ2+4	INH	Alarm detection inhibit	0 to $FFFF_H$  TRKF (0: Without tracking) (1: With tracking)	—	BIN 16Bit	0_H	S/U
	+12 +13	MV	Manipulated value	-10 to 110	%	Real number	0.0	S/U
	+14 +15	SV	Set value	RL to RH	—	Real number	0.0	U
	+16 +17	DV	Deviation	(-110 to 110)	%	Real number	0.0	S
	+18 +19	HS0	Hysteresis	0 to 999999	—	Real number	0.0	U
	+22 +23	RH	Engineering value upper limit	-999999 to 999999	—	Real number	100.0	U
	+24 +25	RL	Engineering value lower limit	-999999 to 999999	—	Real number	0.0	U
	+46 +47	CT	Control cycle	0 to 999999 Note that $\frac{CT}{\Delta T} \leqq 32767$	s	Real number	1.0	U
Loop tag past value memory [*2] [*3]	+96	—	—	Used by the system as a work area.	—	—	—	S
	+97							
Set value [*4]	Ⓢ3+0 +1	E2	Set value	-10 to 110	%	Real number	0.0	S/U

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

*2 The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

*3 The applications of the loop tag past value memory are indicated below.

Specified position	Description
Ⓓ2+96	Control cycle counter initial preset flag
+97	Control cycle counter[*5]

When control is to be started from the initial status, the data must be cleared with the sequence program.

*4 The set value (E2) becomes valid when the set value pattern (SVPTN) is set to "E2 is used".
When using the MV of the upper loop as the set value (E2), specify the device where the manipulated value (MV) of the upper loop is set (offset + 12: MV).
When not using E2 as the set value, make sure to specify a dummy device.
(Special register SD1506 can be specified as a dummy device.)

*5 The control cycle counter rounds off the data to the nearest whole number.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) SV setting processing

Either of the following processings is performed depending on the operation mode (MODE) setting.

(a) When the operation mode (MODE) is any of CAS, CCB and CSV

1) When the set value (E2) is specified, engineering value conversion is performed with the following expression and then "(2) Tracking processing" is performed.

$$SV_n=\frac{RH-RL}{100}\times E2+RL$$

2) When the set value (E2) is not specified, "(2) Tracking processing" is performed without the engineering value conversion being performed.

(b) When the operation mode (MODE) is any of MAN, AUT, CMV, CMB, CAB, LCM, LCA and LCC, "(2) Tracking processing" is performed.

(2) Tracking processing

(a) The set value (SV) is converted reversely from the engineering value with the following operation expression to calculate SVn'.

$$SV_n'=\frac{100}{RH-RL}\times (SV_n-RL)$$

(b) When all of the following conditions hold, tracking processing is performed.

1) The tracking bit (TRK) of the operation constant is 1.

2) The set value (E2) is used.

3) The operation mode (MODE) is any of MAN, AUT, CMV, CMB, CAB, LCM, LCA and LCC.
E2 = SVn'

(c) When the set value (E2) is the manipulated value (MV) of the upper loop, the tracking flag (TRKF) of the alarm detection inhibition (INH) in the upper loop turns to 1.

(3) MV compensation

After the deviation (DV) is calculated from the input value (E1) and the set value after tracking processing (SVn'), the MV compensation value (MV') is calculated.

(a) Calculation of deviation (DV)

The deviation (DV) is calculated under the following condition.

Condition	DV
Forward operation (PN = 1)	E1 - SVn'
Reverse operation (PN = 0)	SVn' - E1

(b) Calculation of MV compensation value (MV')

The MV compensation value (MV')is calculated under the following condition.

Condition	MV'
DV ≥ HS0	100%
DV ≤ - HS0	0%
- HS0 < DV < HS0	Last value (BW value)

(4) MV output

The manipulated value (MV(BW)) is calculated under the following condition.

Condition	BW
CMV, MAN, CMB, LCM	BW = MVn

CSV, CCB, CAB, CAS, AUT, LCC, LCA	BW = MV'
	MVn = BW

(5) 2-position ON/OFF control

BB1 of BB is output under the following condition.

Condition	BB1
|BW| ≥ 50%	1
|BW| < 50%	0

(6) Loop stop processing

(a) Setting 1 in SPA of the alarm detection (ALM) selects a loop stop.
A loop stop performs the following processing and terminates the S.ONF2 instruction.

1) BW and BB1 retain the last values.

2) The operation mode (MODE) is changed to MAN.

(b) Setting 0 in SPA of the alarm detection (ALM) selects a loop run.
A loop run performs "(7) Control cycle judgment".

(7) Control cycle judgment

(a) If the specified control cycle is not reached

1) When the operation mode (MODE) is any of CSV, CCB, CAB, CAS, AUT, LCC and LCA, BW is retained and the S.ONF2 instruction is terminated.

2) When the operation mode (MODE) is any of MAN, CMB, CMV and LCM, BW is made equal to MV and the processing of "(5) 2-position ON/OFF control" is performed.

(b) If the specified control cycle is reached, "(1) SV setting processing" is performed.

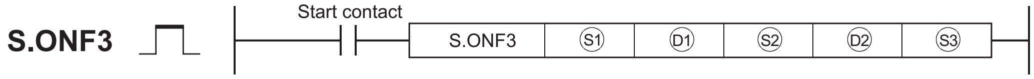
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

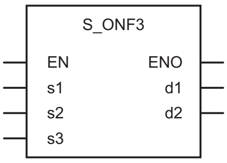
Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of S1, D2 are either a non-numeric or non-normalized number	○	○	—
	When CT < 0 or the execution cycle (SD1500) < 0	○	○	○
	When HS0 < -HS0	○	○	○
4140	When the values of S1, D2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

9.21 S.ONF3

Ladder diagram



Structured ladder/FBD



Structured text language

ENO:=S_ONF3(EN,s1,s2,s3,d1,d2);

Input argument	EN	: Execution condition	: Bit
	Ⓢ1	: Input data start device	: Real data type
	Ⓢ2	: Operation constant start device	: Array of any 16-bit data (0..2)
	Ⓢ3	: When set value (E2) is used: Set value start device When set value (E2) is not used: Dummy device[*1]	: Real data type
Output argument	ENO	: Execution result	: Bit
	Ⓓ1	: Block memory start device	: Array of any 16-bit data (0..2)
	Ⓓ2	: Loop tag memory start device	: Array of any 16-bit data (0..127)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
Ⓢ1	—	○		—					
Ⓓ1	—	○		—					
Ⓢ2	—	○		—					
Ⓓ2	—	○		—					
Ⓢ3	—	○		—					

[*1] Special register SD1506 can be specified as a dummy device.

Function

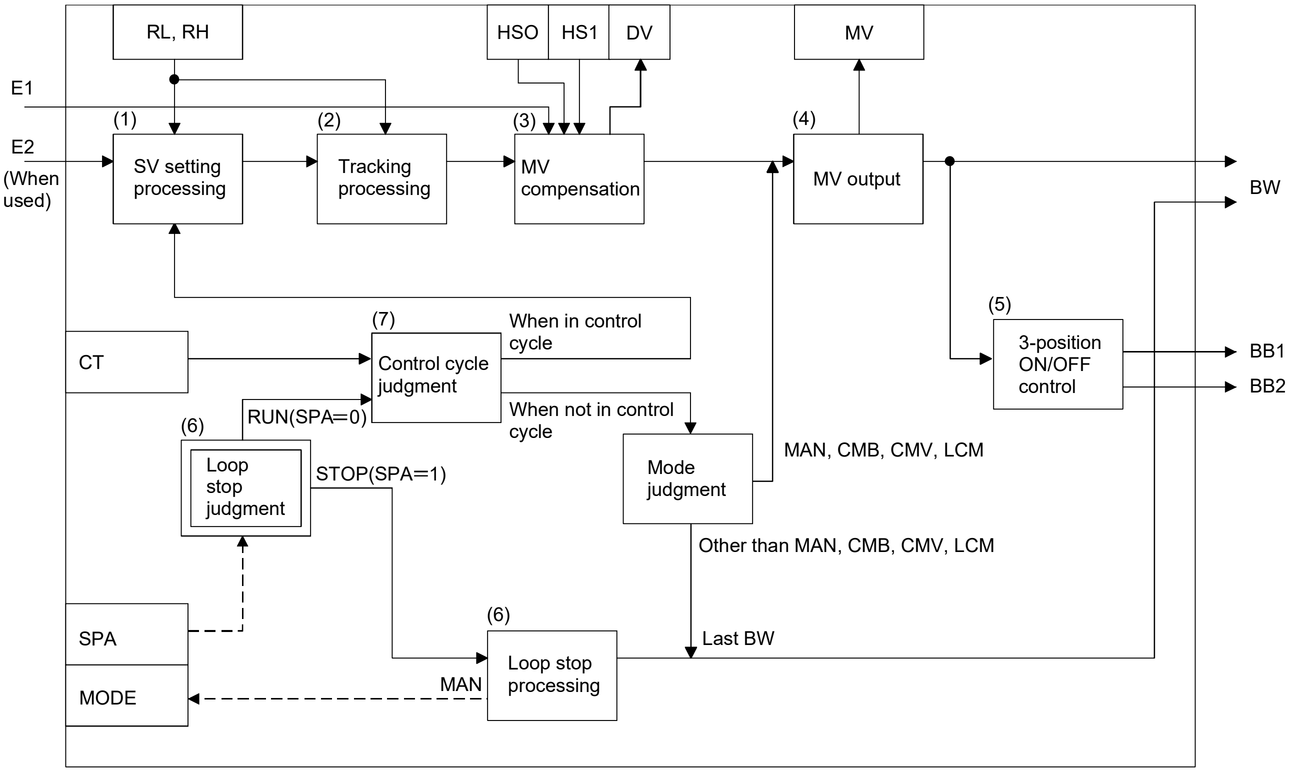
Performs 3-position ON/OFF control (ON/OFF of two contact) when the specified control cycle is reached.

Also performs SV setting processing, tracking processing, MV compensation and MV output processing at this time.

Block diagram

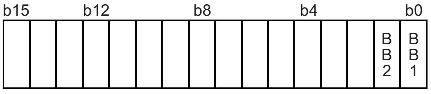
The processing block diagram of the S.ONF3 instruction is shown below.

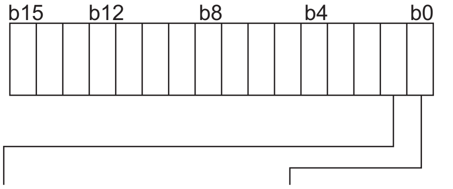
(The numerals (1) to (7) in the diagram indicate the order of the processing.)

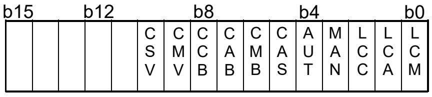
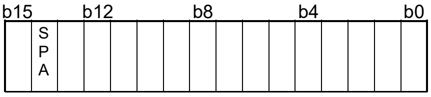


Set Data

(1) Data specified in S.ONF3 instruction

<table>
<tr><th colspan="2">Specified position</th><th>Symbol</th><th>Name</th><th>Recommended range [*1]</th><th>Unit</th><th>Data format</th><th>Standard value</th><th>Set by</th></tr>
<tr><td>Input data</td><td>S1+0
+1</td><td>E1</td><td>Input value</td><td>-999999 to 999999</td><td>%</td><td>Real number</td><td>—</td><td>U</td></tr>
<tr><td rowspan="4">Block memory</td><td>D1+0
+1</td><td>BW</td><td>Output value</td><td>(-999999 to 999999)</td><td>%</td><td>Real number</td><td>—</td><td>S</td></tr>
<tr><td rowspan="3">+2</td><td>BB</td><td colspan="6">—</td></tr>
<tr><td>BB1</td><td>Operation result</td><td rowspan="2">

<table>
<tr><th>Output value</th><th>BB2</th><th>BB1</th></tr>
<tr><td>BW<25%</td><td>1</td><td>0</td></tr>
<tr><td>25%≤BW<75%</td><td>0</td><td>0</td></tr>
<tr><td>BW≥75%</td><td>0</td><td>1</td></tr>
</table></td><td rowspan="2">—</td><td rowspan="2">BIN 16bit</td><td rowspan="2">—</td><td rowspan="2">S</td></tr>
<tr><td>BB2</td><td>Operation result</td></tr>
<tr><td rowspan="3">Operation constant</td><td>S2+0</td><td>PN</td><td>Operation mode</td><td>0: Reverse operation
1: Forward operation</td><td>—</td><td>BIN 16bit</td><td>0</td><td>U</td></tr>
<tr><td>+1</td><td>TRK</td><td>Tracking bit</td><td>0: Without tracking
1: With tracking</td><td>—</td><td>BIN 16bit</td><td>0</td><td>U</td></tr>
<tr><td>+2</td><td>SVPTN</td><td>Set value pattern</td><td>0 to 3


Set value pattern[*3]
0: E2 is upper loop MV.
1: E2 is not upper loop MV.
Set value used[*2]
0: E2 is used.
1: E2 is not used.</td><td>—</td><td>BIN 16bit</td><td>3</td><td>U</td></tr>
<tr><td rowspan="2">Loop tag memory [*4]</td><td>D2+1</td><td>MODE</td><td>Operation mode</td><td>0 to FFFF$_H$

</td><td>—</td><td>BIN 16bit</td><td>8$_H$</td><td>S/U</td></tr>
<tr><td>+3</td><td>ALM</td><td>Alarm detection</td><td>0 to FFFF$_H$


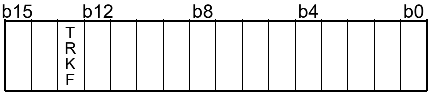
SPA
0: Loop RUN
1: Loop STOP</td><td>—</td><td>BIN 16bit</td><td>4000$_H$</td><td>S/U</td></tr>
</table>

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[*2] Specify whether the set value (E2) is to be used or not.

[*3] Specify whether the MV of the upper loop is to be used or not as the set value (E2).

[*4] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Loop tag memory [*2]	Ⓓ2+4	INH	Alarm detection inhibit	0 to FFFF$_H$  TRKF (0: Without tracking) (1: With tracking)	—	BIN 16Bit	0$_H$	S/U
	+12 +13	MV	Manipulated value	-10 to 110	%	Real number	0.0	S/U
	+14 +15	SV	Set value	RL to RH	—	Real number	0.0	U
	+16 +17	DV	Deviation	(-110 to 110)	%	Real number	0.0	S
	+18 +19	HS0	Hysteresis 0	0 to 999999	—	Real number	0.0	U
	+20 +21	HS1	Hysteresis 1	0 to 999999	—	Real number	0.0	U
	+22 +23	RH	Engineering value upper limit	-999999 to 999999	—	Real number	100.0	U
	+24 +25	RL	Engineering value lower limit	-999999 to 999999	—	Real number	0.0	U
	+46 +47	CT	Control cycle	0 to 999999 Note that $\frac{CT}{\Delta T} \leqq 32767$	s	Real number	1.0	U
Loop tag past value memory [*2] [*3]	Ⓓ2+96 +97	—	—	Used by the system as a work area.	—	—	—	S
Set value [*4]	Ⓢ3+0 +1	E2	Set value	-10 to 110	%	Real number	0.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[*2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

[*3] The applications of the loop tag past value memory are indicated below.

Specified position	Description
Ⓓ2+96	Control cycle counter initial preset flag
+97	Control cycle counter[*5]

When control is to be started from the initial status, the data must be cleared with the sequence program.

[*4] The set value (E2) becomes valid when the set value pattern (SVPTN) is set to "E2 is used".
When using the MV of the upper loop as the set value (E2), specify the device where the manipulated value (MV) of the upper loop is set (offset + 12: MV).
When not using E2 as the set value, make sure to specify a dummy device.
(Special register SD1506 can be specified as a dummy device.)

[*5] The control cycle counter rounds off the data to the nearest whole number.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) SV setting processing

Either of the following processings is performed depending on the operation mode (MODE) setting.

(a) When the operation mode (MODE) is any of CAS, CCB and CSV

1) When the set value (E2) is specified, engineering value conversion is performed with the following expression and then "(2) Tracking processing" is performed.

$$SV_n=\frac{RH-RL}{100}\times E2+RL$$

2) When the set value (E2) is not specified, "(2) Tracking processing" is performed without the engineering value conversion being performed.

(b) When the operation mode (MODE) is any of MAN, AUT, CMV, CMB, CAB, LCM, LCA and LCC, "(2) Tracking processing" is performed.

(2) Tracking processing

(a) The set value (SV) is converted reversely from the engineering value with the following operation expression to calculate SVn'.

$$SV_n'=\frac{100}{RH-RL}\times(SV_n-RL)$$

(b) When all of the following conditions hold, tracking processing is performed.

1) The tracking bit (TRK) of the operation constant is 1.

2) The set value (E2) is used.

3) The operation mode (MODE) is any of MAN, AUT, CMV, CMB, CAB, LCM, LCA and LCC.
E2 = SVn'

(c) When the set value (E2) is the manipulated value (MV) of the upper loop, the tracking flag (TRKF) of the alarm detection inhibition (INH) in the upper loop turns to 1.

(3) MV compensation

After the deviation (DV) is calculated from the input value (E1) and the set value after tracking processing (SVn'), the MV compensation value (MV') is calculated.

(a) Calculation of deviation (DV)

The deviation (DV) is calculated under the following condition.

Condition	DV
Forward operation (PN = 1)	E1 - SVn'
Reverse operation (PN = 0)	SVn' - E1

(b) Calculation of MV compensation value (MV')

The MV compensation value (MV')is calculated under the following condition.

Condition	MV'
DV ≥ (HS1 + HS0)	100%
DV ≤ - (HS1 + HS0)	0%
(- HS1 + HS0) < DV < (HS1 - HS0)	50%
Other than above	Last value (BW value)

(4) MV output

The manipulated value (MV(BW)) is calculated under the following condition.

Condition	BW
CMV, MAN, CMB, LCM	BW = MVn

CSV, CCB, CAB, CAS, AUT, LCC, LCA	BW = MV'
	MVn = BW

(5) 3-position ON/OFF control

BB1 and BB2 of BB are output under the following condition.

Condition	BB1	BB2
BW ≥ 75%	1	0
25% ≤ BW < 75%	0	0
BW < 25%	0	1

(6) Loop stop processing

(a) Setting 1 in SPA of the alarm detection (ALM) selects a loop stop.
A loop stop performs the following processing and terminates the S.ONF3 instruction.

1) BW, BB1 and BB2 retain the last values.

2) The operation mode (MODE) is changed to MAN.

(b) Setting 0 in SPA of the alarm detection (ALM) selects a loop run.
A loop run performs "(7) Control cycle judgment".

(7) Control cycle judgment

(a) If the specified control cycle is not reached

1) When the operation mode (MODE) is any of CSV, CCB, CAB, CAS, AUT, LCC and LCA, BW is retained and the S.ONF3 instruction is terminated.

2) When the operation mode (MODE) is any of MAN, CMB, CMV and LCM, BW is made equal to MV and the processing of "(5) 3-position ON/OFF control" is performed.

(b) If the specified control cycle is reached, "(1) SV setting processing" is performed.

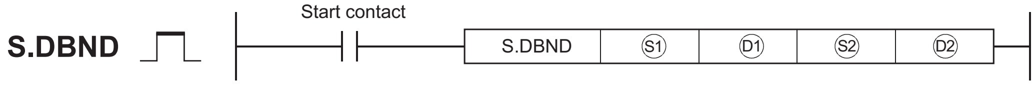
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

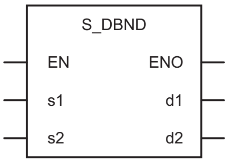
Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When HS0 < 0	○	○	—
	When the values of Ⓢ1, Ⓓ2 are either a non-numeric or non-normalized number	○	○	—
	When CT < 0 or the execution cycle (SD1500) < 0	○	○	○
	When (HS1 + HS0) < 0	○	○	○
	When HS1 < 0	○	○	○
4140	When the values of Ⓢ1, Ⓓ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○
	When HS0 < 0	—	—	○

9.22 S.DBND

Ladder diagram



Structured ladder/FBD



Structured text language

ENO:=S_DBND(EN,s1,s2,d1,d2);

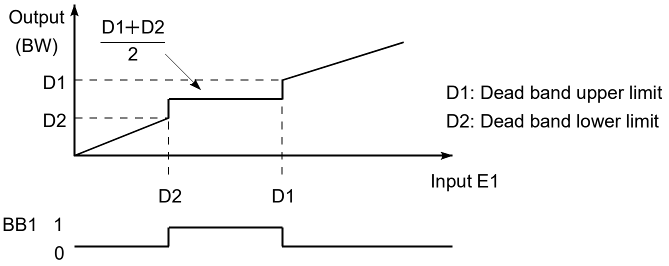
Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Real data type
	S2	: Operation constant start device	: Array of real data type (0..1)
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

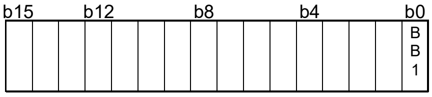
Function

Provides a dead band and performs output processing.



Set Data

(1) Data specified in S.DBND instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value	-999999 to 999999	%	Real number	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	%	Real number	—	S
	+2	BB	—					
		BB1	Dead band action	 (0: Outside the dead band range) (1: Within the dead band range)	—	BIN 16bit	—	S
Operation constant	Ⓢ2+0 +1	D1	Dead band upper limit	-999999 to 999999	—	Real number	100.0	U
	+2 +3	D2	Dead band lower limit	-999999 to 999999	—	Real number	0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) The S.DBND instruction performs the following processing.

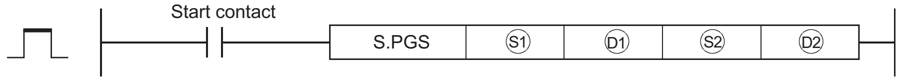
Condition	BW	BB1
D2 ≤ E1 ≤ D1	$\frac{D2+D1}{2}$	1
(E1 < D2) or (E1 > D1)	E1	0

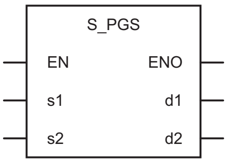
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	○	○	—
4140	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

9.23 S.PGS

Ladder diagram	
S.PGS	

Structured ladder/FBD	Structured text language
	ENO:=S_PGS(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	S1	: Dummy device	: Real data type
	S2	: Dummy device*1	: Real data type
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Loop tag memory start device	: Array of any 16-bit data (0..127)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

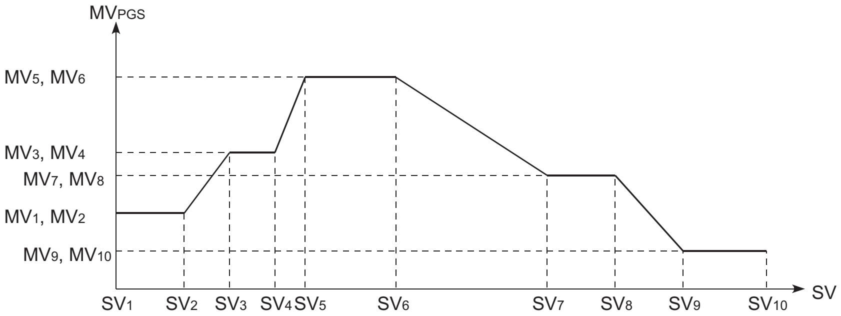
*1 Special register SD1506 can be specified as a dummy device.

Function

Provides a control output according to the SV and MV pattern.

As the output types of the S.PGS instruction, there are three types of the "hold type", "return type" and "cyclic type".

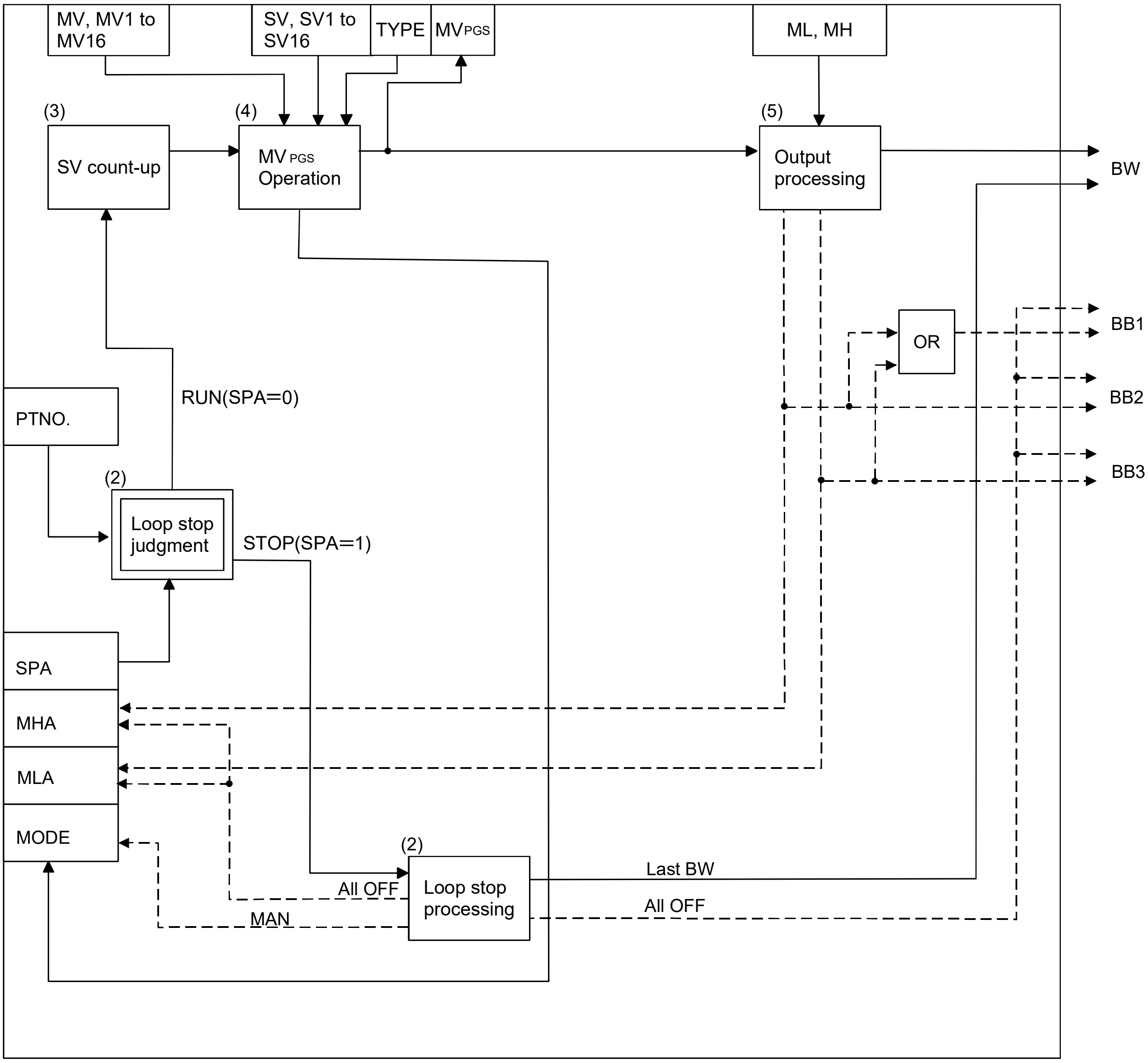
- Hold type : Output is provided with the SV10 value held.
- Return type : The set value (SV) is set to 0 and the last value is output as the manipulated value (MV).
- Cyclic type : After SV1 to SV10 have been processed, processing is restarted from SV1 and the output is provided.



Block diagram

The processing block diagram of the S.PGS instruction is shown below.

(The numerals (2) to (5) in the diagram indicate the order of the processing.)



Set Data

(1) Data specified in S.PGS instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Block memory	D1+0 +1	BW	Output value	(-999999 to 999999)	%	Real number	—	S
	+2	BB	—					
		BB1	Alarm	b15 b12 b8 b4 b0 BB3 BB2 BB1 (0: Without alarm) (1: With alarm)	—	BIN 16bit	—	S
		BB2	Output upper limit alarm					
		BB3	Output lower limit alarm					
Loop tag memory [*2]	D2+1	MODE	Operation mode	0 to FFFF$_H$ b15 b12 b8 b4 b0 CSV CMV CCB CAB CMB CAS AUT MAN LCC LCA LCM	—	BIN 16bit	8_H	S/U
	+3	ALM	Alarm detection	0 to FFFF$_H$ b15 b12 b8 b4 b0 SPA MHA MLA SPA 0: Loop RUN 1: Loop STOP MHA, MLA (0: Without alarm) (1: With alarm)	—	BIN 16bit	4000_H	S/U
	+4	INH	Alarm detection inhibition	0 to FFFF$_H$ b15 b12 b8 b4 b0 ERRI MHI MLI 0: Alarm enable 1: Alarm inhibit	—	BIN 16bit	0_H	S/U
	+10	PTNO	Number of operation constant polygon points	0 to 16	—	BIN 16bit	0	U
	+12 +13	MV	Manipulated value	-10 to 110	%	Real number	0.0	S/U
	+14 +15	SV	Set value	0 to 999999	s	Real number	0.0	S/U
	+16	TYPE	Operation type	0: Hold type operation (When operation mode is AUT or CAB) 1: Return type operation (When operation mode is AUT or CAB)	—	BIN 16bit	0	U
	+18 +19	MH	Output upper limit value	-10 to 110	%	Real number	100.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[*2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Loop tag memory [*2]	+20 +21	ML	Output lower limit value	-10 to 110	%	Real number	0.0	U
	+22 +23	SV1	Setting time 1	0 to 999999	s	Real number	0.0	U
	⋮	⋮	⋮					
	+52 +53	SV16	Setting time 16					
	+54 +55	MV1	Setting output 1	-10 to 110	%	Real number	0.0	U
	⋮	⋮	⋮					
	+84 +85	MV16	Setting output 16					

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[*2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

(2) Execution cycle (∆T)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) Output type

The output type is determined by the combination of the operation mode (MODE) and operation type (TYPE) as indicated below.

Operation mode (MODE)	Operation type (TYPE)	Operation
MAN, CMB, CMV, LCM, LCA, LCC	—	Operation stopped at current SV and MV
AUT, CAB	0	Hold type operation
	1	Return type operation
CAS, CCB, CSV	—	Cyclic type operation

(2) Loop stop processing

(a) Setting 1 in SPA of the alarm detection (ALM) selects a loop stop.

A loop stop performs the following processing and terminates the S.PGS instruction.

1) BW retains the last value.

2) MHA and MLA of the alarm detection (ALM) are turned to 0.

3) The operation mode (MODE) is changed to MAN.

4) BB1 to BB3 of BB are turned to 0.

(b) Setting 0 in SPA of the alarm detection (ALM) selects a loop run.

A loop run performs "(3) SV count-up processing".

(3) SV count-up processing

SV count-up is performed with the following expression in each execution cycle.

SV' = SV + ΔT

(4) MVPGS operation

Type		Hold	Return	Cyclic
Mode		AUT, CAB		CAS, CCB, CSV
MVPGS operation	$SV < SV_1$	MV_1		
	$SV_{n-1} \leq SV < SV_n$	$\frac{MV_n - MV_{n-1}}{SV_n - SV_{n-1}} \times (SV' - SV_{n-1}) + MV_{n-1}$		
Processing when SV' > SV_n	Mode change	MAN	MAN	Not moved
	SV	Last value	0	0
	MV	Last value	Last value	MV_1
	Restart method	After SV is set, mode is changed from MAN to AUT.	Mode is changed from MAN to AUT.	Automatic restart

(5) Output processing

Condition	Manual			Automatic		
	MAN, CMB, CMV, LCM, LCA, LCC			AUT, CAB, CAS, CCB, CSV		
	BW	BB2, MHA	BB3, MLA	BW	BB2, MHA	BB3, MLA
MVPGS > MH	MV_n	0	0	MV_n = MH	1 [*1]	0
MVPGS < ML	MV_n	0	0	MV_n = ML	0	1 [*2]
Others	MV_n	0	0	MV_n = MVPGS	0	0

[*1] When MHI or ERRI in the alarm detection inhibition (INH) is set to 1, MHA and BB2 show 0 since the alarm is prohibited.

[*2] When MLI or ERRI in the alarm detection inhibition (INH) is set to 1, MLA and BB3 show 0 since the alarm is prohibited.

Operation Error

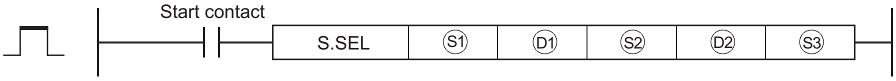
In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

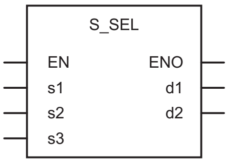
Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the value of D2 is either a non-numeric or non-normalized number	○	○	—
	When PTNO < 0 or 16 < PTNO	○	○	○
4140	When the value of D2 is either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

9.24 S. SEL

Ladder diagram

S.SEL



Structured ladder/FBD	Structured text language
	ENO:=S_SEL(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	(S1)	: Input data 1 start device	: Real data type
	(S2)	: Operation constant start device	: Array of any 16-bit data (0..5)
	(S3)	: Input data 2 start device	: Real data type
Output argument	ENO	: Execution result	: Bit
	(D1)	: Block memory start device	: Array of any 16-bit data (0..2)
	(D2)	: Loop tag memory start device	: Array of any 16-bit data (0..127)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
(S1)	—	○		—					
(D1)	—	○		—					
(S2)	—	○		—					
(D2)	—	○		—					
(S3)	—	○		—					

Function

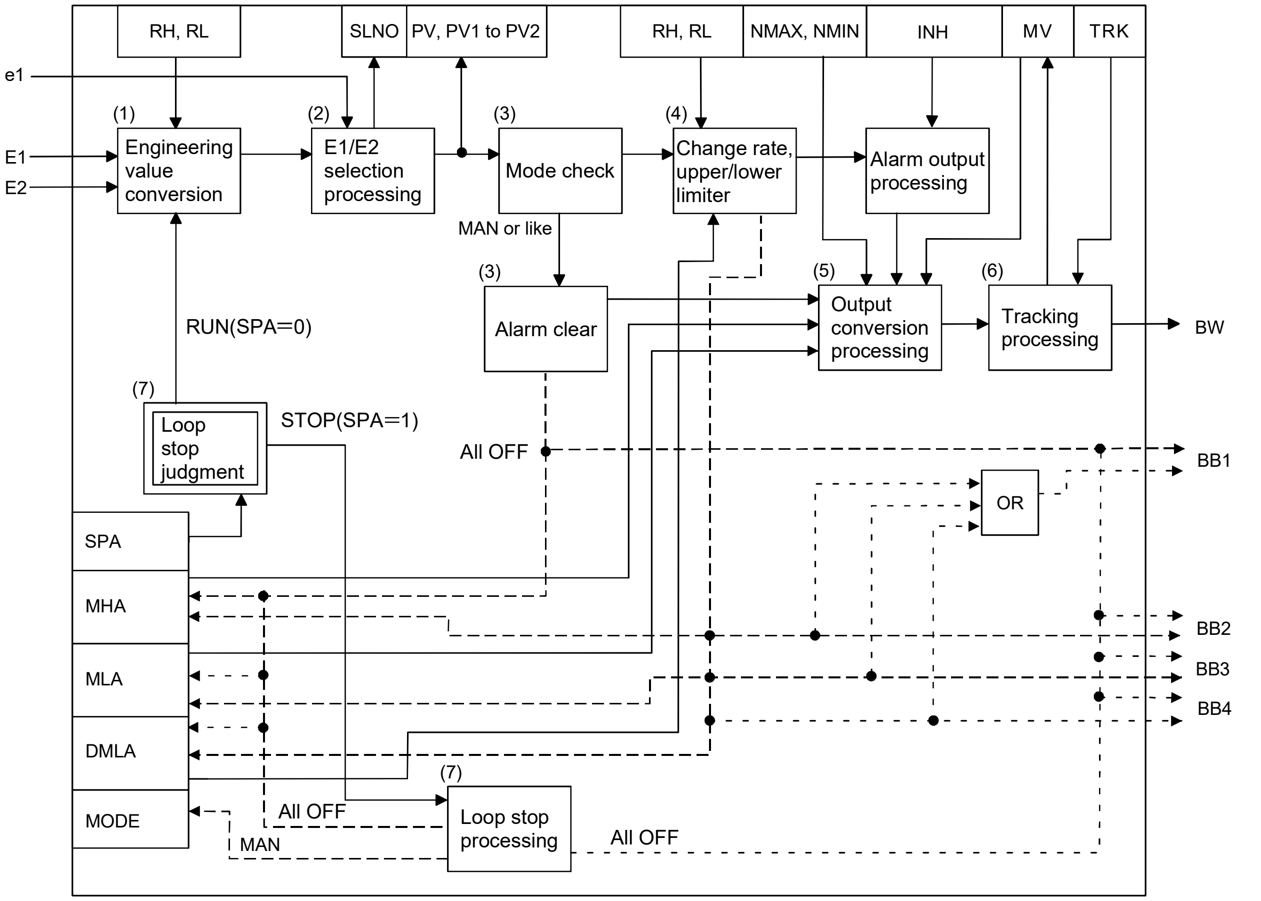
Provides an output in the specified mode (automatic mode/manual mode).

- In the automatic mode, the input value 1 (E1) or input value 2 (E2) selected by the selection signal (e1) is output.
- In the manual mode, the manipulated value (MV) is output.

Block diagram

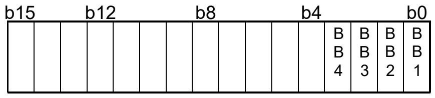
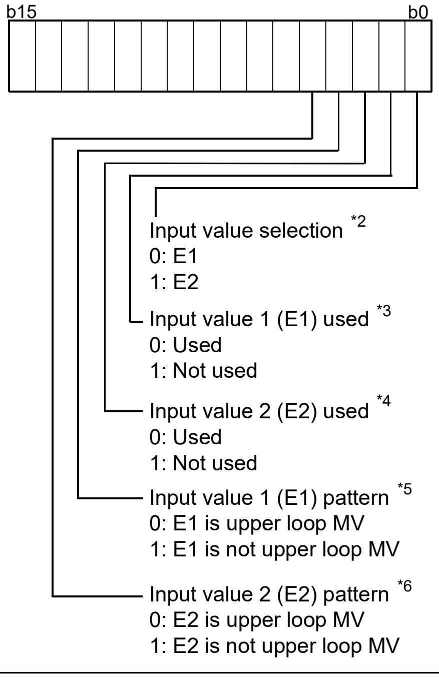
The processing block diagram of the S. SEL instruction is shown below.

(The numerals (1) to (7) in the diagram indicate the order of the processing.)



Set Data

(1) Data specified in S. SEL instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data 1	Ⓢ1+0 +1	E1	Input value 1	-999999 to 999999	%	Real number	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	—	Real number	—	S
	+2	BB	—					
		BB1	Alarm	 (0: Without alarm) (1: With alarm)	—	BIN 16bit	—	S
		BB2	Output upper limit alarm					
		BB3	Output lower limit alarm					
		BB4	Output change rate alarm					
Operation constant	Ⓢ2+0 +1	NMAX	Output conversion upper limit	-999999 to 999999	—	Real number	100.0	U
	+2 +3	NMIN	Output conversion lower limit	-999999 to 999999	—	Real number	0.0	U
	+4	TRK	Tracking bit	0: Without tracking 1: With tracking	—	BIN 16it	0	U
	+5	SVPTN	Set value pattern	 Input value selection [*2] 0: E1 1: E2 Input value 1 (E1) used [*3] 0: Used 1: Not used Input value 2 (E2) used [*4] 0: Used 1: Not used Input value 1 (E1) pattern [*5] 0: E1 is upper loop MV 1: E1 is not upper loop MV Input value 2 (E2) pattern [*6] 0: E2 is upper loop MV 1: E2 is not upper loop MV	—	BIN 16bit	$1E_H$	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

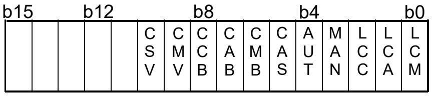
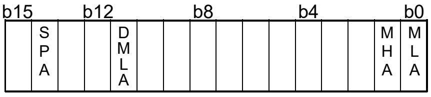
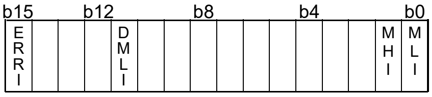
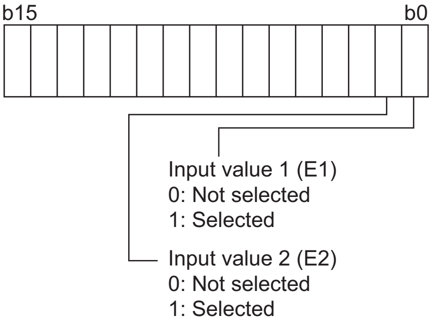
[*2] Select E1 or E2 for the input value.

[*3] Specify whether the input value 1 (E1) is to be used or not.

[*4] Specify whether the input value 2 (E2) is to be used or not.

[*5] Specify whether the MV of the upper loop is to be used or not as the input value 1 (E1).

[*6] Specify whether the MV of the upper loop is to be used or not as the input value 2 (E2).

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Loop tag memory [*2]	Ⓓ2+1	MODE	Operation mode	0 to $FFFF_H$  b15 b12 b8 b4 b0 CSV CMV CCB CAB CMB CAS AUT MAN LCC LCA LCM	—	BIN 16bit	8_H	S/U
	+3	ALM	Alarm detection	0 to $FFFF_H$  SPA 0: Loop RUN 1: Loop STOP DMLA, MHA, MLA (0: Without alarm) (1: With alarm)	—	BIN 16bit	4000_H	S/U
	+4	INH	Alarm detection inhibition	0 to $FFFF_H$  0: Alarm enable 1: Alarm inhibit	—	BIN 16bit	0_H	S/U
	+10 +11	PV	Selection value	RL to RH	—	Real number	0.0	S
	+12 +13	MV	Manipulated value	-10 to 110	%	Real number	0.0	S/U
	+14 +15	PV1	Process value 1	RL to RH	—	Real number	0.0	S
	+16 +17	PV2	Process value 2	RL to RH	—	Real number	0.0	S
	+18 +19	MH	Output upper limit value	-10 to 110	%	Real number	100.0	U
	+20 +21	ML	Output lower limit value	-10 to 110	%	Real number	0.0	U
	+22 +23	RH	Engineering value upper limit	-999999 to 999999	—	Real number	100.0	U
	+24 +25	RL	Engineering value lower limit	-999999 to 999999	—	Real number	0.0	U
	+26	SLNO	Selection No.	 Input value 1 (E1) 0: Not selected 1: Selected Input value 2 (E2) 0: Not selected 1: Selected	—	BIN 16bit	0	S
	+48 +49	DML	Output change rate limit value	0 to 100	%	Real number	100.0	U
Input data 2	Ⓢ3+0 +1	E2	Input value 2	-999999 to 999999	%	Real number	0.0	U

[*1] The data of the item where the recommended range values are indicated within the parentheses is stored by the system. The user cannot set the data.

[*2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Processing contents

(1) Engineering value conversion

Engineering value conversion is performed with the following expression.

$$PV_n=\frac{RH-RL}{100}\times E_n+RL$$

(2) Input value 1 (E1) or input value 2 (E2) selection processing

Whether the input value 1 (E1) or input value 2 (E2) will be used is selected depending on the e1 setting of the set value pattern (SVPTN).

- e1 = 0: Input value 1 (E1) is used PV = PV_1
- e1 = 1: Input value 2 (E2) is used PV = PV_2

SLN0: The bit corresponding to the input value 1 (E1) or input value 2 (E2) is turned to 1.

(3) Mode check

The following processing is performed depending on the operation mode (MODE) setting.

(a) When the operation mode (MODE) is any of MAN, CMB, CMV and LCM

1) "(5) Output conversion processing" is performed.

2) MHA, MLA and DMLA of the alarm detection (ALM) are turned to 0.

3) BB1 to BB4 of BB are turned to 0.

(b) When the operation mode (MODE) is any of AUT, CAB, CAS, CCB, CSV, LCA and LCC

1) Engineering value reverse conversion is performed with the following expression.

$$T=\frac{100}{RH\text{-}RL}\times(PV\text{-}RL)$$

2) "(4) Change rate, upper/lower limiter" is performed.

(4) Change rate, upper/lower limiter

Change rate and upper/lower limit value checks are performed on the input value 1 (E1) or input value 2 (E2).

(a) Change rate limiter

Condition	T'	BB4, DMLA
|T - MVn| ≤ DML	T' = T	0
(T - MVn) > DML	T' = MVn + DML	1 [*1]
(T - MVn) < - DML	T' = MVn - DML	1 [*1]

[*1] When DMLI or ERRI in the alarm detection inhibition (INH) is set to 1, DMLA and BB4 show 0 since the alarm is inhibited.

(b) Upper/lower limiter

Condition	MV	BB2, MHA	BB3, MLA
T' > MH	MVn = MH	1 [*2]	0
T' < ML	MVn = ML	0	1 [*3]
ML ≤ T' ≤ MH	MVn = T'	0	0

[*2] When MHI or ERRI in the alarm detection inhibition (INH) is set to 1, MHA and BB2 show 0 since the alarm is inhibited.

[*3] When MLI or ERRI in the alarm detection inhibition (INH) is set to 1, MLA and BB3 show 0 since the alarm is inhibited.

(5) Output conversion processing

Engineering value conversion is performed with the following expression.

$$BW=\frac{NMAX-NMIN}{100}\times MV_n+NMIN$$

(6) Tracking processing

(a) When all of the following conditions hold, the operation result is output to the input value 1 (E1) or input value 2 (E2).

1) The operation mode (MODE) is any of MAN, CMB, CMV and LCM.

2) The tracking bit (TRK) is 1.

En = MVn

(b) When all of the following conditions hold, the operation result is output to the input value 1 (E1) or input value 2 (E2).

1) The operation mode (MODE) is any of AUT, CAS, CAB, CCB, CSV, LCA and LCC.

2) The tracking bit (TRK) is 1.

3) BB1 of BB is 1

En = MVn

(7) Loop stop processing

(a) Setting 1 in SPA of the alarm detection (ALM) selects a loop stop.

A loop stop performs the following processing and terminates the S.SEL instruction.

1) BW retains the last value.

2) DMLA, MHA and MLA of the alarm detection (ALM) are turned to 0.

3) The operation mode (MODE) is changed to MAN.

4) BB1 to BB4 of BB are turned to 0.

(b) Setting 0 in SPA of the alarm detection (ALM) selects a loop run.

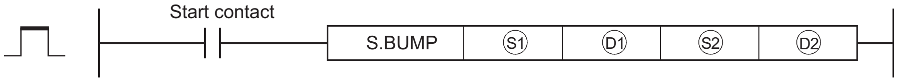
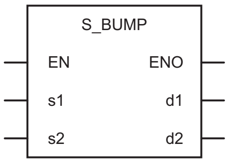
A loop run performs "(1) Engineering value conversion ".

Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of S1, D1, S2, D2 are either a non-numeric or non-normalized number	○	○	—
4140	When the values of S1, D1, S2, D2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

9.25 S.BUMP

Ladder diagram	
S.BUMP 	
Structured ladder/FBD	**Structured text language**
	ENO:=S_BUMP(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Array of any 16-bit data (0..4)
	S2	: Operation constant start device	: Array of real data type (0..1)
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Real data type
	D2	: Local work memory start device	: Array of real data type (0..1)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

Function

Brings the output value (BW) closer to the output set value (E1) from the output control value (E2) at the fixed rate when the operation mode is switched from the manual mode to the automatic mode.

Brings the output value (BW) closer to the output set value (E1) with a primary delay when the output value (BW) falls within the range specified as the delay zone (a) relative to the output set value (E1).

Set Data

(1) Data specified in S. BUMP instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	(S1)+0 +1	E1	Output set value	-999999 to 999999	%	Real number	—	U
	+2 +3	E2	Output control value	-999999 to 999999	%	Real number	—	U
	+4	e1	Mode switching signal	0: Manual mode 1: Automatic mode	—	BIN 16Bit	—	U
Block memory	(D1)+0 +1	BW	Output value	(-999999 to 999999)	%	Real number	—	S
Operation constant	(S2)+0 +1	T	Delay time	0 to 999999	s	Real number	1.0	U
	+2 +3	a	Delay zone	0 to 999999	%	Real number	1.0	U
Local work memory [*2]	(D2)+0 +1	Xq	Initial deviation value	Used by the system as a work area.	—	Real number	1.0	S
	+2 +3	Xp	Deviation					

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

*2 The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) Either of the following processings is performed depending on the mode select signal (e1) setting of the input data.

(a) In the manual mode (e1 = 0), the output value (BW), initial deviation value (Xg) and deviation (Xp) are calculated with the following expressions.

- BW = output control value (E2)
- Xq = output control value (E2) - output set value (E1)
- Xp = output control value (E2) - output set value (E1)

(b) In the automatic mode (e1 = 1), the output value is calculated with the following expression.

Condition	|Xp| > a	|Xp| ≤ a
Xp	$X_p = X_{p'} - \frac{\Delta T}{T} X_q$	$X_p = \frac{T}{T + \Delta T} X_{p'}$
BW	BW = E1 + Xp On the assumption that $\|X_p\| \leqq \frac{\Delta T}{T} \|X_q\|$ • BW = E1 • Xp = Xp'	BW = E1 + Xp On the assumption that $\|X_p\| \leq 10^{-4}$ • BW = E1 • Xp = Xp'

However, when T ≤ ΔT in the automatic mode, BW = E1, Xp = Xp'

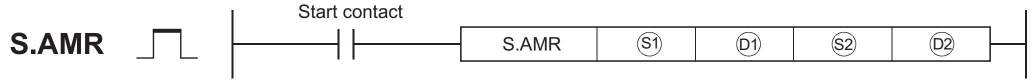
Operation Error

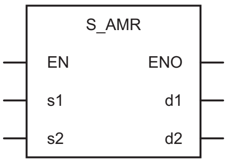
In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	○	○	—
4140	When the values of Ⓢ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

9.26 S.AMR

Ladder diagram



Structured ladder/FBD	Structured text language
	ENO:=S_AMR(EN,s1,s2,d1,d2);

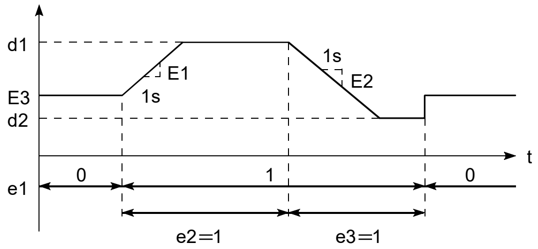
Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Array of any 16-bit data (0..6)
	S2	: Operation constant start device	: Array of real data type (0..1)
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Real data type
	D2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

[*1] Special register SD1506 can be specified as a dummy device.

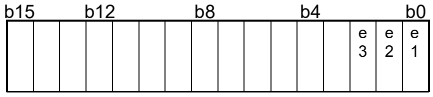
Function

Increases or decreases the output value at the fixed rate.



Set Data

(1) Data specified in S.AMR instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	S1+0 +1	E1	Output addition value	-999999 to 999999	—	Real number	—	U
	+2 +3	E2	Output subtraction value	-999999 to 999999	—	Real number	—	U
	+4 +5	E3	Output set value	-999999 to 999999	—	Real number	—	U
	+6	e1	Operation select signal	 e1 0: Manual mode 1: Automatic mode e2 0: Not added 1: Added e3 0: Not subtracted 1: Subtracted	—	BIN 16Bit	—	U
		e2	Output addition signal					
		e3	Output subtraction signal					
Block memory	D1+0 +1	BW	Output value	(-999999 to 999999)	—	Real number	—	S
Operation constant	S2+0 +1	d1	Output upper limit value	0 to 999999	—	Real number	1.0	U
	+2 +3	d2	Output lower limit value	0 to 999999	—	Real number	1.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) Either of the following processings is performed depending on the settings of the operation select signal (e1), output addition signal (e2) and output subtraction signal (e3).

(a) In the manual mode (e1 = 0), BW = E3.

(b) In the automatic mode (e1 = 1), any of the operations in the following table is performed depending on the settings of the output addition signal (e2) and output subtraction signal (e3).

e2	e3	BW
1	0	BW = BW + |E1| × ΔT On the assumption that d1≤ BW: BW = d1
0	1	BW = BW - |E2| × ΔT On the assumption that BW ≤ d2: BW = d2
1	1	BW = BW
0	0	

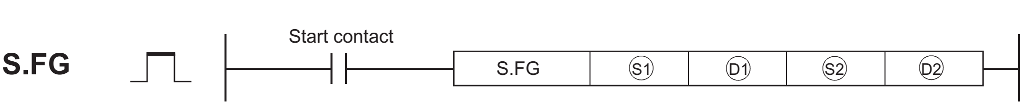
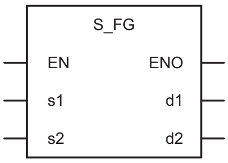
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	○	○	—
4140	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

CHAPTER 10 COMPENSATION OPERATION INSTRUCTIONS

10.1 S.FG

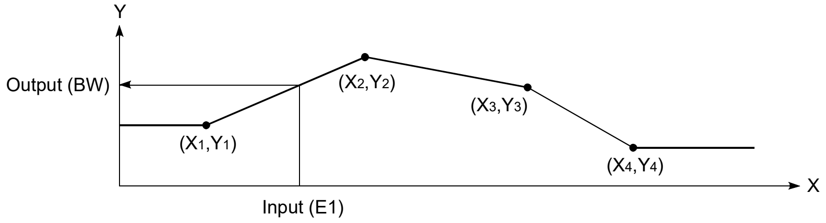
Ladder diagram	
S.FG 	
Structured ladder/FBD	**Structured text language**
	ENO:=S_FG(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Real data type
	S2	: Operation constant start device	: Any 16-bit data
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Real data type
	D2	: Local work memory start device	: Array of real data type (0..95)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

Function

In response to the input value (E1), outputs the value following the function generator pattern that consists of n pieces of polygon points specified as the operation constants.



Set Data

(1) Data specified in S.FG instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	S1+0 +1	E1	Input value	-999999 to 999999	—	Real number	—	U
Block memory	D1+0 +1	BW	Output value	(-999999 to 999999)	—	Real number	—	S
Operation constant	S2+0	SN	Number of polygon points	0 to 48	—	BIN 16Bit	0	U
Local work memory	D2+0 +1	X1	Polygon point coordinates	-999999 to 999999	—	Real number	—	U
	+2 +3	Y1	Polygon point coordinates					
	+4 +5	X2	Polygon point coordinates					
	+6 +7	Y2	Polygon point coordinates					
	⋮	⋮	⋮					
	+4SN-4 +4SN-3	Xn	Polygon point coordinates					
	+4SN-2 +4SN-1	Yn	Polygon point coordinates					

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) The S.FG instruction performs the following operation.

Condition	Output value (BW)
$E1 \leq X1$	BW = Y1
$X_{i-1} < E1 \leq X_i$ (i = 2 to n)	$BW = \frac{Y_i - Y_{i-1}}{X_i - X_{i-1}} \times (E1 - X_{i-1}) + Y_{i-1}$
$Xn < E1$	BW = Yn

(2) When n = 0 there is no processing.

(3) When $X_{i-1} > X_i$, the value is cut off to n = i-1 (Data after that is ignored.)

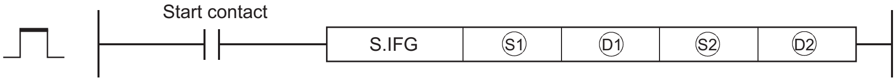
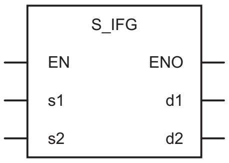
When there are multiple Y_i for the same X_i, the lowest i is selected.

Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of S1, D2 are either a non-numeric or non-normalized number	○	○	—
	When (SN < 0) or (SN > 48)	○	○	○
4140	When the values of S1, D2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

10.2 S.IFG

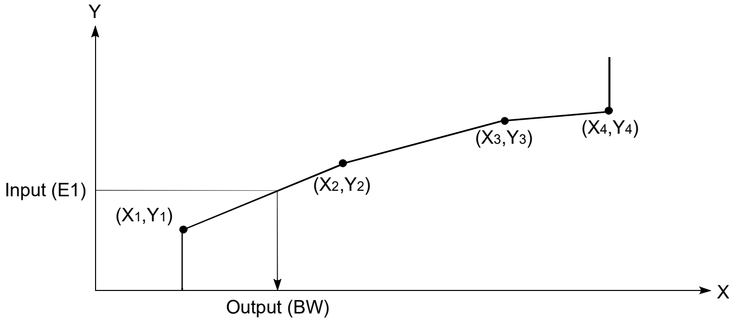
Ladder diagram	
S.IFG 	
Structured ladder/FBD	**Structured text language**
	ENO:=S_IFG(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Real data type
	S2	: Operation constant start device	: Any 16-bit data
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Real data type
	D2	: Local work memory start device	: Array of real data type (0..95)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

Function

In response to the input value (E1), outputs the value following the inverse function generator pattern that consists of n pieces of polygon points specified as the operation constants.



Set Data

(1) Data specified in S.IFG instruction

Specified position		Symbol	Name	Recommended range *1	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value	-999999 to 999999	—	Real number	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	—	Real number	—	S
Operation constant	Ⓢ2+0	SN	Number of polygon points	0 to 48	—	BIN 16Bit	0	U
Local work memory	Ⓓ2+0 +1	X1	Polygon point coordinates	-999999 to 999999	—	Real number	—	U
	+2 +3	Y1	Polygon point coordinates					
	+4 +5	X2	Polygon point coordinates					
	+6 +7	Y2	Polygon point coordinates					
	⋮	⋮	⋮					
	+4SN-4 +4SN-3	Xn	Polygon point coordinates					
	+4SN-2 +4SN-1	Yn	Polygon point coordinates					

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) The S.IFG instruction performs the following operation.

Condition	Output value (BW)
$E1 \leq Y1$	BW=X1
$Y_{i-1} < E1 \leq Y_i$ (i = 2 to n)	$BW = \frac{X_i - X_{i-1}}{Y_i - Y_{i-1}} \times (E1 - Y_{i-1}) + X_{i-1}$
$Yn < E1$	BW=Xn

(2) When n = 0 there is no processing.

(3) When $Y_{i-1} > Y_i$, the value is cut off to n = i-1 (Data after that is ignored.)
When there are multiple X_i for the same Y_i, the lowest i is selected.

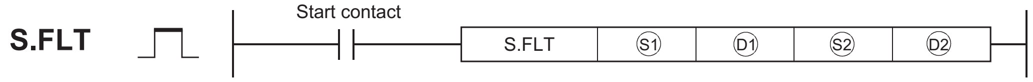
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

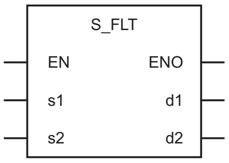
Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓓ2 are either a non-numeric or non-normalized number	○	○	—
	When (SN < 0) or (SN > 48)	○	○	○
4140	When the values of Ⓢ1, Ⓓ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

10.3 S.FLT

Ladder diagram



Structured ladder/FBD



Structured text language

ENO:=S_FLT(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Real data type
	S2	: Operation constant start device	: Array of any 16-bit data (0..2)
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Local work memory start device	: Array of any 16-bit data (0..103)

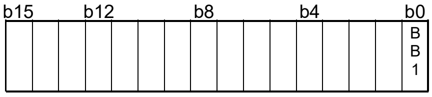
Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

Function

Stores SN pieces of input values (E1) sampled at the data collection intervals (ST) into the dead time table, and outputs the average of those SN pieces of data.

Set Data

(1) Data specified in S.FLT instruction

Specified position		Symbol	Name	Recommended range [1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value	-999999 to 999999	—	Real number	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	—	Real number	—	S
	+2	BB	—					
		BB1	Data sufficiency bit	 (0: Data sufficiency) (1: Data insufficiency)	—	BIN 16Bit	—	S
Operation constant	Ⓢ2+0 +1	ST	Data collection interval	0 to 999999	s	Real number	1.0	U
	+2	SN	Sampling count	0 to 48	—	BIN 16Bit	0	U
Local work memory [2]	Ⓓ2+0 +1	ST'	Last data collection interval	Used by the system as a work area.	—	Real number	—	S
	+2	SN'	Last sampling count		—	BIN 16Bit	—	S
	+3	i	Cycle counter[3]		—	BIN 16Bit	—	S
	+4	n1	Number of stored data		—	BIN 16Bit	—	S
	+5	n2	Store		—	BIN 16Bit	—	S
	+6 +7	—	—	—	—	—	—	—
	+8 +9	1	Dead time table 1	Used by the system as a work area.	—	Real number	—	S
	+10 +11	2	Dead time table 2					
	⋮	⋮	⋮					
	+2SN+6 +2SN+7	SN	Dead time table SN					

[1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

[3] The cycle counter rounds off the data to the nearest whole number.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) The data update cycle is $\frac{ST}{\Delta T}$. (The decimal is rounded down.)

(2) The data sufficiency bit (BB1) turns to 0 when the dead time table is filled with SN pieces of data.
It turns to 1 when the dead time table is not filled.

Point

- Until the dead time table is filled with data, the average of the data provided until then is output.
- Processed using ST = n × ΔT. (n is an integral)

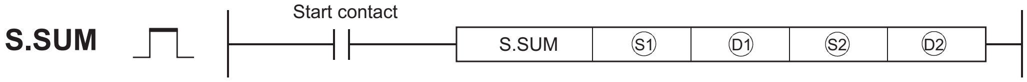
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

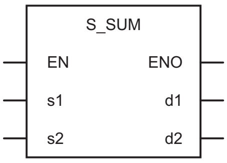
Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	○	○	—
	When (SN < 0) or (SN > 48)	○	○	○
	When the execution cycle < 0	○	○	○
	When ST < 0	○	○	○
4140	When the values of Ⓢ1, Ⓢ2, Ⓓ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

10.4 S.SUM

Ladder diagram



Structured ladder/FBD



Structured text language

ENO:=S_SUM(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	Ⓢ1	: Input data start device	: Array of any 16-bit data (0..2)
	Ⓢ2	: Operation constant start device	: Array of any 16-bit data (0..4)
Output argument	ENO	: Execution result	: Bit
	Ⓓ1	: Block memory start device	: Real data type
	Ⓓ2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
Ⓢ1	—	○		—					
Ⓓ1	—	○		—					
Ⓢ2	—	○		—					
Ⓓ2	—	○		—					

[*1] Special register SD1506 can be specified as a dummy device.

Function

Integrates and outputs the input value (E1) when the integration start signal (e1) turns from 0 to 1.

Set Data

(1) Data specified in S.SUM instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value	-999999 to 999999	—	Real number	—	U
	+2	e	—					
		e1	Integration start signal	b15 b12 b8 b4 b0 e1 0: Integration not executed 1: Integration executed	—	BIN 16Bit	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	—	Real number	—	S
Operation constant	Ⓢ2+0 +1	ILC	Input low cut value	-999999 to 999999	—	Real number	0.0	U
	+2 +3	A	Initial value	-999999 to 999999	—	Real number	0.0	U
	+4	RANGE	Input range	1: /Second 2: /Minute 3: /Hour	—	BIN 16Bit	1	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as real numbers.

Processing contents

(1) The S.SUM instruction performs the following processing.

e1	E1	Output (BW)
0	—	The initial value (A) of the operation constant is output.
1	E1 ≤ ILC	The last value is output unchanged.
	E1 > ILC	$BW = E1 \times \frac{\Delta T}{T} + \text{Last value}$

(2) The T value used for the operation changes depending on the input range (RANGE) setting.

- When RANGE = 1, T = 1
- When RANGE = 2, T = 60
- When RANGE = 3, T = 3600

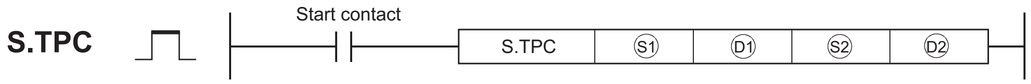
Operation Error

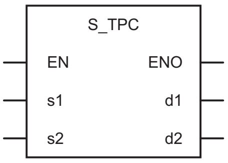
In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓓ1, Ⓢ2 are either a non-numeric or non-normalized number	○	○	—
	When the RANGE setting is other than 1 to 3	○	○	○
4140	When the values of Ⓢ1, Ⓓ1, Ⓢ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

10.5 S.TPC

Ladder diagram



Structured ladder/FBD	Structured text language
	ENO:=S_TPC(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Array of any 16-bit data (0..6)
	S2	: Operation constant start device	: Array of real data type (0..4)
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Real data type
	D2	: Dummy device[*1]	: Any 16-bit data

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

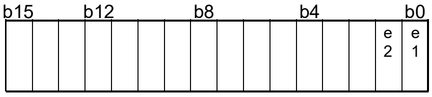
*1 Special register SD1506 can be specified as a dummy device.

Function

The input value (E1) is subject to temperature/pressure correction (temperature or pressure) and output.

Set Data

(1) Data specified in S.TPC instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Differential pressure	-999999 to 999999	—	Real number	—	U
	+2 +3	E2	Measurement temperature	-999999 to 999999	—	Real number	—	U
	+4 +5	E3	Measured pressure	-999999 to 999999	—	Real number	—	U
	+6	e	—					
		e1	E2 use flag	 0: Unused 1: Used	—	BIN 16Bit	—	U
		e2	E3 use flag					
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	—	Real number	—	S
Operation constant	Ⓢ2+0 +1	TEMP	Design temperatureT' (Engineering value)	-999999 to 999999	°C	Real number	0.0	U
	+2 +3	B1	Bias (Temperature)	-999999 to 999999	°C	Real number	273.15	U
	+4 +5	PRES	Design pressureP' (Engineering value)	-999999 to 999999	—	Real number	0.0	U
	+6 +7	B2	Bias (Pressure)	-999999 to 999999	—	Real number	10332.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) The S.TPC instruction calculates the temperature/pressure correction value with the following expression.
BW = E1 × A1 × A2

(2) A1 and A2 use the values calculated with the following expressions.

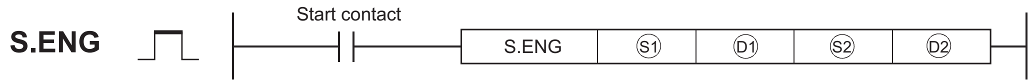
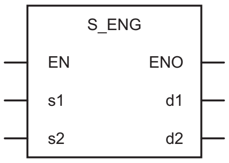
Input		A1	A2
E2	E3		
Used	Used	$\frac{T'+B1}{E2+B1}$	$\frac{E3+B2}{P'+B2}$
Not used	Used	1.0	$\frac{E3+B2}{P'+B2}$
Used	Not used	$\frac{T'+B1}{E2+B1}$	1.0
Not used	Not used	1.0	1.0

Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of (S1), (S2) are either a non-numeric or non-normalized number	○	○	—
4140	When the values of (S1), (S2) are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

10.6 S.ENG

Ladder diagram	
	
Structured ladder/FBD	**Structured text language**
	ENO:=S_ENG(EN,s1,s2,d1,d2);

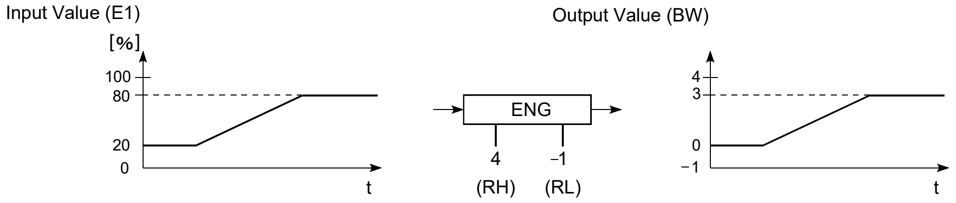
Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Real data type
	S2	: Operation constant start device	: Array of real data type (0..1)
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Real data type
	D2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

Function

The input value (E1) is output by the engineering conversion.



Set Data

(1) Data specified in S.ENG instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	S1+0 +1	E1	Input value	-999999 to 999999	%	Real number	—	U
Block memory	D1+0 +1	BW	Output value	(-999999 to 999999)	—	Real number	—	S
Operation constant	S2+0 +1	RH	Engineering value upper limit	-999999 to 999999	—	Real number	100.0	U
	+2 +3	RL	Engineering value lower limit	-999999 to 999999	—	Real number	0.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) The S.ENG instruction performs the following operation.

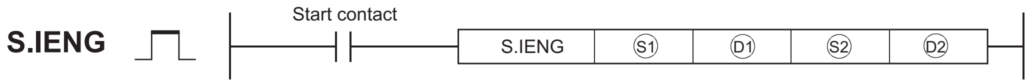
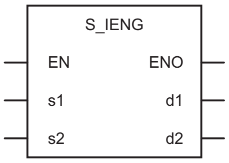
$$BW = \frac{RH-RL}{100} \times E1 + RL \quad (E1 = 0 \text{ to } 100\%)$$

Operation Error

In the following cases, the error flag (SM) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of S1, S2 are either a non-numeric or non-normalized number	○	○	—
4140	When the values of S1, S2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

10.7 S.IENG

Ladder diagram	
	
Structured ladder/FBD	**Structured text language**
	ENO:=S_IENG(EN,s1,s2,d1,d2);

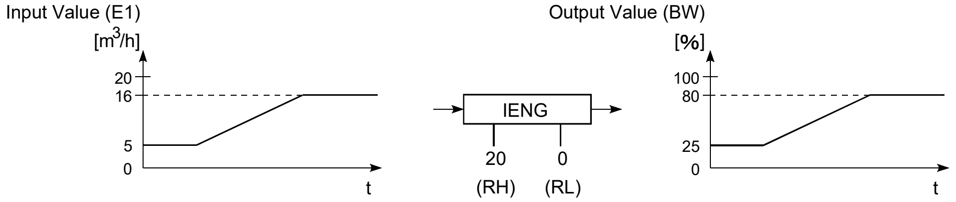
Input argument	EN	: Execution condition	: Bit
	Ⓢ1	: Input data start device	: Real data type
	Ⓢ2	: Operation constant start device	: Array of real data type (0..1)
Output argument	ENO	: Execution result	: Bit
	Ⓓ1	: Block memory start device	: Real data type
	Ⓓ2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
Ⓢ1	—	○		—					
Ⓓ1	—	○		—					
Ⓢ2	—	○		—					
Ⓓ2	—	○		—					

[*1] Special register SD1506 can be specified as a dummy device.

Function

The input value (E1) is converted to % value and output.



Set Data

(1) Data specified in S.IENG instruction

Specified Position		Symbol	Name	Recommended range *1	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value	-999999 to 999999	—	Real number	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	%	Real number	—	S
Operation constant	Ⓢ2+0 +1	RH	Engineering value upper limit	-999999 to 999999	—	Real number	100.0	U
	+2 +3	RL	Engineering value lower limit	-999999 to 999999	—	Real number	0.0	U

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) The S.IENG instruction performs the following operation.

$$BW = \frac{100}{RH - RL} \times (E1 - RL) \quad (\%)$$

(2) Make setting to satisfy RH > RL.

(3) If RH ≤ RL, the processing is executed unchanged but engineering value reverse conversion is not performed.

(4) If RH = RL, BW = 0.

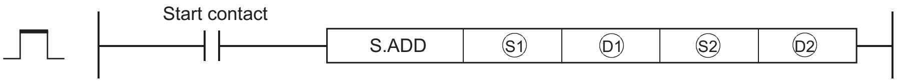
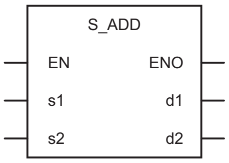
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	○	○	—
4140	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

CHAPTER 11 ARITHMETIC OPERATION INSTRUCTIONS

11.1 S.ADD

Ladder diagram	
S.ADD 	
Structured ladder/FBD	**Structured text language**
	ENO:=S_ADD(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	Ⓢ1	: Input data start device	: Array of any 16-bit data (0..10)
	Ⓢ2	: Operation constant start device	: Array of any 16-bit data (0..12)
Output argument	ENO	: Execution result	: Bit
	Ⓓ1	: Block memory start device	: Real data type
	Ⓓ2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
Ⓢ1	—	○		—					
Ⓓ1	—	○		—					
Ⓢ2	—	○		—					
Ⓓ2	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

Function

The input value (El to En) data is added by attaching a coefficient.

Set Data

(1) Data specified in S.ADD instruction

Specification position		Symbol	Name	Recommended range[*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0	n	Input count	0 to 5	—	BIN 16Bit	—	U
	+1 +2	E1	Input value 1	-999999 to 999999	—	Real number	—	U
	+3 +4	E2	Input value 2					
	⋮	⋮	⋮					
	+2n-1 +2n	En	Input value n					
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	—	Real number	—	S
Operation constant	Ⓢ2+0	n	Number of coefficients	0 to 5	—	BIN 16Bit	0	U
	+1 +2	K1	Coefficient 1	-999999 to 999999	—	Real number	1.0	U
	+3 +4	K2	Coefficient 2					
	⋮	⋮	⋮					
	+2n-1 +2n	Kn	Coefficient n					
	+2n+1 +2n+2	B	Bias	-999999 to 999999	—	Real number	0.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) The S.ADD instruction performs the following operation.
BW = (K1 × E1) + (K2 × E2) … + (Kn × En) + B

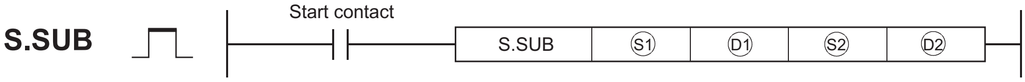
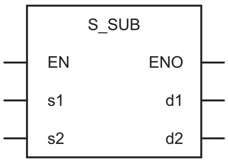
(2) When n is 0, BW = B.

Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	○	○	—
	When not n = 0 to 5	○	○	○
4140	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

11.2 S.SUB

<table>
<tr><th colspan="2">Ladder diagram</th></tr>
<tr><td colspan="2"> </td></tr>
<tr><th>Structured ladder/FBD</th><th>Structured text language</th></tr>
<tr><td> </td><td>ENO:=S_SUB(EN,s1,s2,d1,d2);</td></tr>
</table>

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Array of any 16-bit data (0..10)
	S2	: Operation constant start device	: Array of any 16-bit data (0..12)
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Real data type
	D2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

Function

The input value (E1 to En) data is subtracted by attaching a coefficient.

Set Data

(1) Data specified in S.SUB instruction

Specification position		Symbol	Name	Recommended range[*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0	n	Input count	0 to 5	—	BIN 16Bit	—	U
	+1 +2	E1	Input value 1	-999999 to 999999	—	Real number	—	U
	+3 +4	E2	Input value 2					
	⋮	⋮	⋮					
	+2n-1 +2n	En	Input value n					
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	—	Real number	—	S
Operation constant	Ⓢ2+0	n	Number of coefficients	0 to 5	—	BIN 16Bit	0	U
	+1 +2	K1	Coefficient 1	-999999 to 999999	—	Real number	1.0	U
	+3 +4	K2	Coefficient 2					
	⋮	⋮	⋮					
	+2n-1 +2n	Kn	Coefficient n					
	+2n+1 +2n+2	B	Bias	-999999 to 999999	—	Real number	0.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) The S.SUB instruction instructs the following operation.
BW = (K1 × E1) - (K2 × E2) ... - (Kn × En) + B

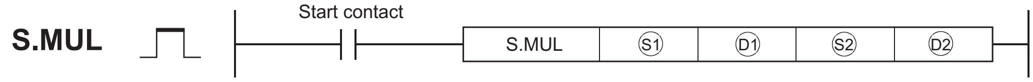
(2) When n is 0, BW = B.

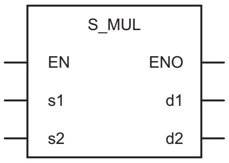
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	○	○	—
	When not n = 0 to 5	○	○	○
4140	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

11.3 S.MUL

Ladder diagram
S.MUL 

Structured ladder/FBD	Structured text language
	ENO:=S_MUL(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	Ⓢ1	: Input data start device	: Array of any 16-bit data (0..10)
	Ⓢ2	: Operation constant start device	: Array of any 16-bit data (0..12)
Output argument	ENO	: Execution result	: Bit
	Ⓓ1	: Block memory start device	: Real data type
	Ⓓ2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
Ⓢ1	—	○		—					
Ⓓ1	—	○		—					
Ⓢ2	—	○		—					
Ⓓ2	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

Function

The input value (E1 to En) data is multiplied by attaching a coefficient.

Set Data

(1) Data specified in S.MUL instruction

Specification position		Symbol	Name	Recommended range[*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0	n	Input count	0 to 5	—	BIN 16Bit	—	U
	+1 +2	E1	Input value 1	-999999 to 999999	—	Real number	—	U
	+3 +4	E2	Input value 2					
	⋮	⋮	⋮					
	+2n-1 +2n	En	Input value n					
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	—	Real number	—	S
Operation constant	Ⓢ2+0	n	Number of coefficients	0 to 5	—	BIN 16Bit	0	U
	+1 +2	K1	Coefficient 1	-999999 to 999999	—	Real number	1.0	U
	+3 +4	K2	Coefficient 2					
	⋮	⋮	⋮					
	+2n-1 +2n	Kn	Coefficient n					
	+2n+1 +2n+2	B	Bias	-999999 to 999999	—	Real number	0.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) The S.MUL instruction performs the following operation.
BW = (K1 × E1) × (K2 × E2) ... × (Kn × En) + B

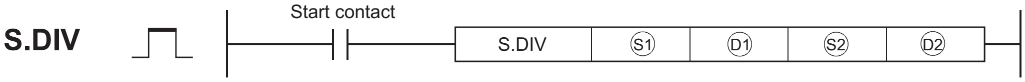
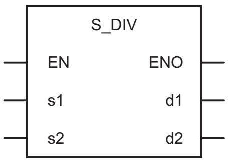
(2) When n is 0, BW = B.

Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	○	○	—
	When not n = 0 to 5	○	○	○
4140	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

11.4 S.DIV

<table>
<tr><th colspan="2">Ladder diagram</th></tr>
<tr><td colspan="2">
</td></tr>
<tr><th>Structured ladder/FBD</th><th>Structured text language</th></tr>
<tr><td>
</td><td>ENO:=S_DIV(EN,s1,s2,d1,d2);</td></tr>
</table>

Input argument	EN	: Execution condition	: Bit
	Ⓢ1	: Input data start device	: Array of real data type (0..1)
	Ⓢ2	: Operation constant start device	: Array of real data type (0..5)
Output argument	ENO	: Execution result	: Bit
	Ⓓ1	: Block memory start device	: Real data type
	Ⓓ2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
Ⓢ1	—	○		—					
Ⓓ1	—	○		—					
Ⓢ2	—	○		—					
Ⓓ2	—	○		—					

[*1] Special register SD1506 can be specified as a dummy device.

Function

Divides the input value 1 (E1) by the input value 2 (E2).

Set Data

(1) Data specified in S.DIV instruction

Specification position		Symbol	Name	Recommended range[*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value 1 (Minute)	-999999 to 999999	—	Real number	—	U
	+2 +3	E2	Input value 2 (Denominator)	-999999 to 999999	—	Real number	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(-999999 to 999999)	—	Real number	—	S
Operation constant	Ⓢ2+0 +1	A	Coefficient 1	-999999 to 999999	—	Real number	1.0	U
	+2 +3	K1	Coefficient 2	-999999 to 999999	—	Real number	1.0	U
	+4 +5	K2	Coefficient 3	-999999 to 999999	—	Real number	1.0	U
	+6 +7	B1	Bias 1	-999999 to 999999	—	Real number	0.0	U
	+8 +9	B2	Bias 2	-999999 to 999999	—	Real number	0.0	U
	+10 +11	B3	Bias 3	-999999 to 999999	—	Real number	0.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) The S.DIV instruction performs the following operation.

$$BW = A \times \frac{K1 \times E1 + B1}{K2 \times E2 + B2} + B3$$

(2) When the denominator is 0, BW = B3.

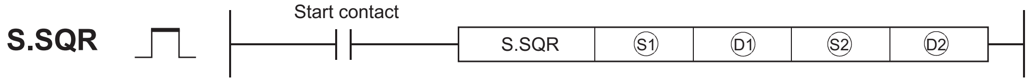
Operation Error

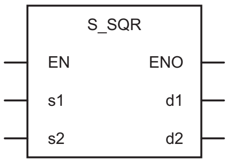
In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	○	○	—
4140	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

11.5 S.SQR

Ladder diagram



Structured ladder/FBD	Structured text language
	ENO:=S_SQR(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	(s1)	: Input data start device	: Real data type
	(s2)	: Operation constant start device	: Array of real data type (0..1)
Output argument	ENO	: Execution result	: Bit
	(d1)	: Block memory start device	: Real data type
	(d2)	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
(s1)	—	○		—					
(d1)	—	○		—					
(s2)	—	○		—					
(d2)	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

Function

The √ of input value (EI) is output.

When the input value is negative, 0 is output.

Set Data

(1) Data specified in S.SQR instruction

Specification position		Symbol	Name	Recommended range[*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value	0 to 999999	—	Real number	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(0 to 999999)	—	Real number	—	S
Operation constant	Ⓢ2+0 +1	OLC	Output low cut value	0 to 999999	—	Real number	0.0	U
	+2 +3	K	Coefficient	0 to 999999	—	Real number	10.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) The S.SQR instruction performs the following operation.

$BW = K \times \sqrt{(E1)}$

(2) When $K \times \sqrt{(E1)} \leqq 0LC$, BW = 0.
Also, when (E1 < 0), BW = 0.

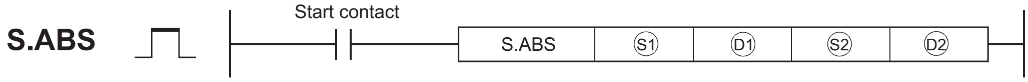
Operation Error

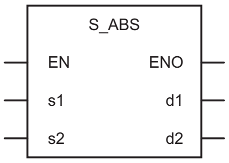
In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	○	○	—
4140	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

11.6 S.ABS

Ladder diagram



Structured ladder/FBD	Structured text language
	ENO:=S_ABS(EN,s1,s2,d1,d2);

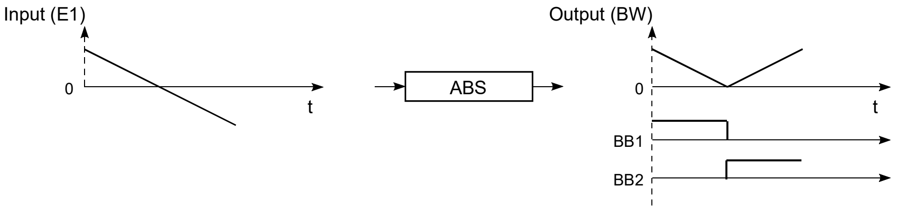
Input argument	EN	: Execution condition	: Bit
	Ⓢ1	: Input data start device	: Real data type
	Ⓢ2	: Dummy device[*1]	: Real data type
Output argument	ENO	: Execution result	: Bit
	Ⓓ1	: Block memory start device	: Array of any 16-bit data (0..2)
	Ⓓ2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
Ⓢ1	—	○		—					
Ⓓ1	—	○		—					
Ⓢ2	—	○		—					
Ⓓ2	—	○		—					

*1 Special register SD1506 can be specified as a dummy device.

Function

The input value (E1) absolute value is output.



Set Data

(1) Data specified in S.ABS instruction

Specification position		Symbol	Name	Recommended range[*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value	-999999 to 999999	—	Real number	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(0 to 999999)	—	Real number	—	S
	+2	BB	—					
		BB1	Judgment of input value (E1) sign	b15 b12 b8 b4 b0 BB2 (b1), BB1 (b0) When E1＞0: BB1＝1 When E1＜0: BB2＝1 When E1＝0: BB1＝BB2＝0	—	BIN 16Bit	—	S
		BB2						

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) The S.ABS instruction performs the following operation.
BW = |E1|

(2) The sign of the input value 1 (E1) is judged and the result is output to BB1 and BB2.

E1 Status	BB1	BB2
E1 > 0	1	0
E1 < 0	0	1
E1 = 0	0	0

Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

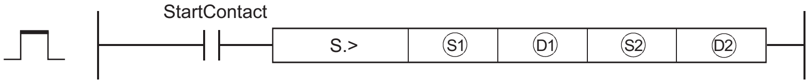
Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When the value of Ⓢ1 is either a non-numeric or non-normalized number	○	○	—
4140	When the value of Ⓢ1 is either a non-numeric or non-normalized number	—	—	○

CHAPTER 12 COMPARISON OPERATION INSTRUCTIONS

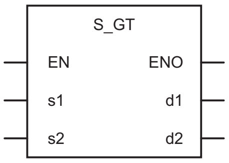
12.1 S. >

Ladder diagram

S.>



Structured ladder/FBD



Structured text language

ENO:=S_GT(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Array of real data type (0..1)
	S2	: Operation constant start device	: Array of real data type (0..1)
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

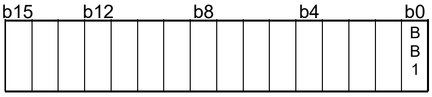
[*1] Special register SD1506 can be specified as a dummy device.

Function

Compares the input value 1 (E1) and input value 2 (E2) and outputs the result of comparison.

Set Data

(1) Data specified in S. > instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	S1+0 +1	E1	Input value 1	-999999 to 999999	—	Real number	—	U
	+2 +3	E2	Input value 2	-999999 to 999999	—	Real number	—	U
Block memory	D1+0 +1	BW	Output value	(The same value as the input value 1 (E1) is stored)	—	Real number	—	S
	+2	BB	—					
		BB1	Comparison output	 (The result of comparison between E1 and E2 is stored.)	—	BIN 16Bit	—	S
Operation constant	S2+0 +1	K	Set value	-999999 to 999999	—	Real number	0.0	U
	+2 +3	HS	Hysteresis	0 to 999999	—	Real number	0.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) Compares the input value 1 (E1) with the input value 2 (E2), and outputs the result of the comparison to BB1 of the block memory.

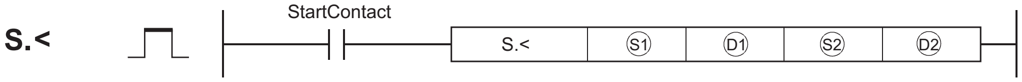
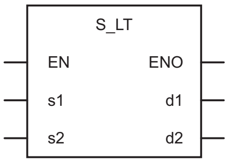
Condition	BB1
E1 > (E2 + K)	1
E1 ≤ (E2 + K - HS)	0
(E2 + K - HS) < E1 ≤ (E2 + K)	Last value is output

Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of S1, S2 are either a non-numeric or non-normalized number	○	○	—
	When the hysteresis value is negative	○	○	—
4140	When the values of S1, S2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○
	When the hysteresis value is negative	—	—	○

12.2 S. <

Ladder diagram	
	
Structured ladder/FBD	**Structured text language**
	ENO:=S_LT(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Array of real data type (0..1)
	S2	: Operation constant start device	: Array of real data type (0..1)
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

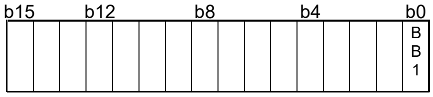
*1 Special register SD1506 can be specified as a dummy device.

Function

Compares the input value 1 (E1) and input value 2 (E2) and outputs the result of comparison.

Set Data

(1) Data specified in S. < instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value 1	-999999 to 999999	—	Real number	—	U
	+2 +3	E2	Input value 2	-999999 to 999999	—	Real number	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(The same value as the input value 1 (E1) is stored)	—	Real number	—	S
	+2	BB	—					
		BB1	Comparison output	 (The result of comparison between E1 and E2 is stored.)	—	BIN 16Bit	—	S
Operation constant	Ⓢ2+0 +1	K	Set value	-999999 to 999999	—	Real number	0.0	U
	+2 +3	HS	Hysteresis	0 to 999999	—	Real number	0.0	U

*1 The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) Compares the input value 1 (E1) with the input value 2 (E2), and outputs the result of the comparison to BB1 of the block memory.

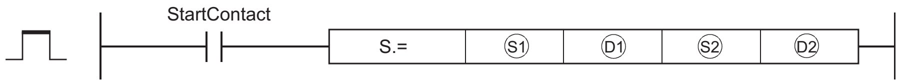
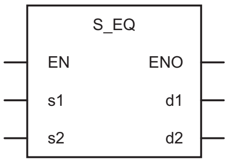
Condition	BB1
E1 < (E2 + K)	1
E1 ≥ (E2 + K + HS)	0
(E2 + K) ≤ E1 < (E2 + K + HS)	Last value is output

Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	○	○	—
	When the hysteresis value is negative	○	○	—
4140	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○
	When the hysteresis value is negative	—	—	○

12.3 S. =

<table>
<tr><th colspan="2">Ladder diagram</th></tr>
<tr><td colspan="2">S.= 
</td></tr>
<tr><th>Structured ladder/FBD</th><th>Structured text language</th></tr>
<tr><td>
</td><td>ENO:=S_EQ(EN,s1,s2,d1,d2);</td></tr>
</table>

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Array of real data type (0..1)
	S2	: Operation constant start device	: Array of real data type (0..1)
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

[*1] Special register SD1506 can be specified as a dummy device.

Function

Compares the input value 1 (E1) and input value 2 (E2) and outputs the result of comparison.

Set Data

(1) Data specified in S. = instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value 1	-999999 to 999999	—	Real number	—	U
	+2 +3	E2	Input value 2	-999999 to 999999	—	Real number	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(The same value as the input value 1 (E1) is stored)	—	Real number	—	S
	+2	BB	—					
		BB1	Comparison output	b15 b12 b8 b4 b0 BB1 (b0) (The result of comparison between E1 and E2 is stored.)	—	BIN 16Bit	—	S
Operation constant	Ⓢ2+0 +1	K	Set value	-999999 to 999999	—	Real number	0.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) Compares the input value 1 (E1) with the input value 2 (E2), and outputs the result of the comparison to BB1 of the block memory.

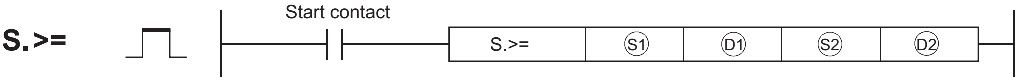
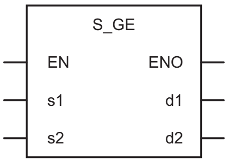
Condition	BB1
E1 = (E2 + K)	1
E1 ≠ (E2 + K)	0

Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	○	○	—
4140	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

12.4 S. >=

Ladder diagram	
S.>= 	
Structured ladder/FBD	**Structured text language**
	ENO:=S_GE(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	(S1)	: Input data start device	: Array of real data type (0..1)
	(S2)	: Operation constant start device	: Array of real data type (0..1)
Output argument	ENO	: Execution result	: Bit
	(D1)	: Block memory start device	: Array of any 16-bit data (0..2)
	(D2)	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
(S1)	—	○		—					
(D1)	—	○		—					
(S2)	—	○		—					
(D2)	—	○		—					

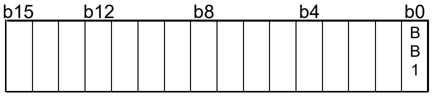
*1 Special register SD1506 can be specified as a dummy device.

Function

Compares the input value 1 (E1) and input value 2 (E2) and outputs the result of comparison.

Set Data

(1) Data specified in S. >= instruction

Specification position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	Ⓢ1+0 +1	E1	Input value 1	-999999 to 999999	—	Real number	—	U
	+2 +3	E2	Input value 2	-999999 to 999999	—	Real number	—	U
Block memory	Ⓓ1+0 +1	BW	Output value	(The same value as the input value 1 (E1) is stored)	—	Real number	—	S
	+2	BB	—					
		BB1	Comparison output	 (The result of comparison between E1 and E2 is stored.)	—	BIN 16Bit	—	S
Operation constant	Ⓢ2+0 +1	K	Set value	-999999 to 999999	—	Real number	0.0	U
	+2 +3	HS	Hysteresis	0 to 999999	—	Real number	0.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) Compares the input value 1 (E1) with the input value 2 (E2), and outputs the result of the comparison to BB1 of the block memory.

Condition	BB1
E1 ≥ (E2 + K)	1
E1 < (E2 + K - HS)	0
(E2 + K - HS) ≤ E1 < (E2 + K)	Last value is output

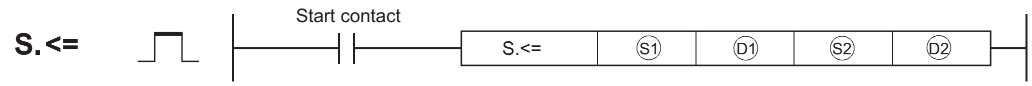
Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

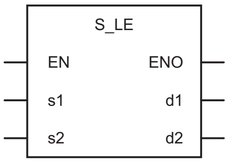
Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	○	○	—
	When the hysteresis value is negative	○	○	—
4140	When the values of Ⓢ1, Ⓢ2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○
	When the hysteresis value is negative	—	—	○

12.5 S. <=

Ladder diagram



Structured ladder/FBD



Structured text language

ENO:=S_LE(EN,s1,s2,d1,d2);

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Array of real data type (0..1)
	S2	: Operation constant start device	: Array of real data type (0..1)
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Array of any 16-bit data (0..2)
	D2	: Dummy device[*1]	: Real data type

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					

[*1] Special register SD1506 can be specified as a dummy device.

Function

Compares the input value 1 (E1) and input value 2 (E2) and outputs the result of comparison.

Set Data

(1) Data specified in S. <= instruction

Specification position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	S1+0 +1	E1	Input value 1	-999999 to 999999	—	Real number	—	U
	+2 +3	E2	Input value 2	-999999 to 999999	—	Real number	—	U
Block memory	D1+0 +1	BW	Output value	(The same value as the input value 1 (E1) is stored)	—	Real number	—	S
	+2	BB	—					
		BB1	Comparison output	b15 b12 b8 b4 b0 BB1 (b0) (The result of comparison between E1 and E2 is stored.)	—	BIN 16Bit	—	S
Operation constant	S2+0 +1	K	Set value	-999999 to 999999	—	Real number	0.0	U
	+2 +3	HS	Hysteresis	0 to 999999	—	Real number	0.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

Processing contents

(1) Compares the input value 1 (E1) with the input value 2 (E2), and outputs the result of the comparison to BB1 of the block memory.

Condition	BB1
E1 ≤ (E2 + K)	1
E1 > (E2 + K + HS)	0
(E2 + K) < E1 ≤ (E2 + K + HS)	Last value is output

Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the values of S1, S2 are either a non-numeric or non-normalized number	○	○	—
	When the hysteresis value is negative	○	○	—
4140	When the values of S1, S2 are either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○
	When the hysteresis value is negative	—	—	○

CHAPTER 13 AUTO TUNING

Auto tuning is designed to make the initial setting of the PID constants.

The auto tuning of the CPU module can be used for processes that can be approximated with a primary delay plus dead time represented by the following expression.

Example) Process with relatively slow response such as temperature adjustment

$$\frac{K}{1+Ts}e^{-Ls}$$

K: Gain, T: Time constant, L: Dead time, s: Laplace operator

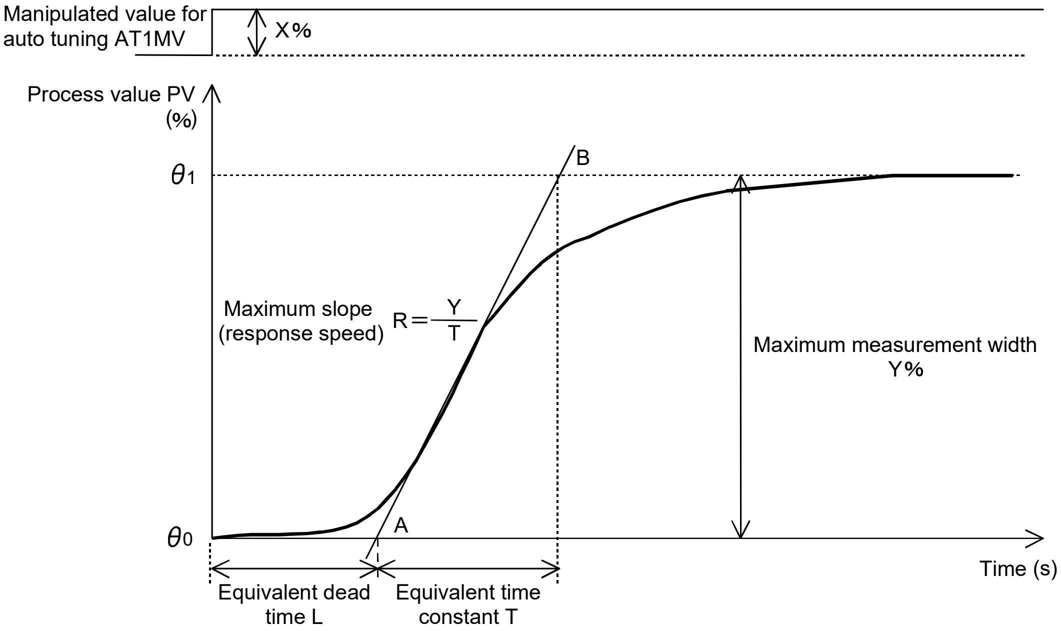
Auto tuning can be used for the loop that uses the S. PID or S. 2PID instruction.

Auto tuning is performed in the ZN process: stepped response process of Ziegler and Nichols.

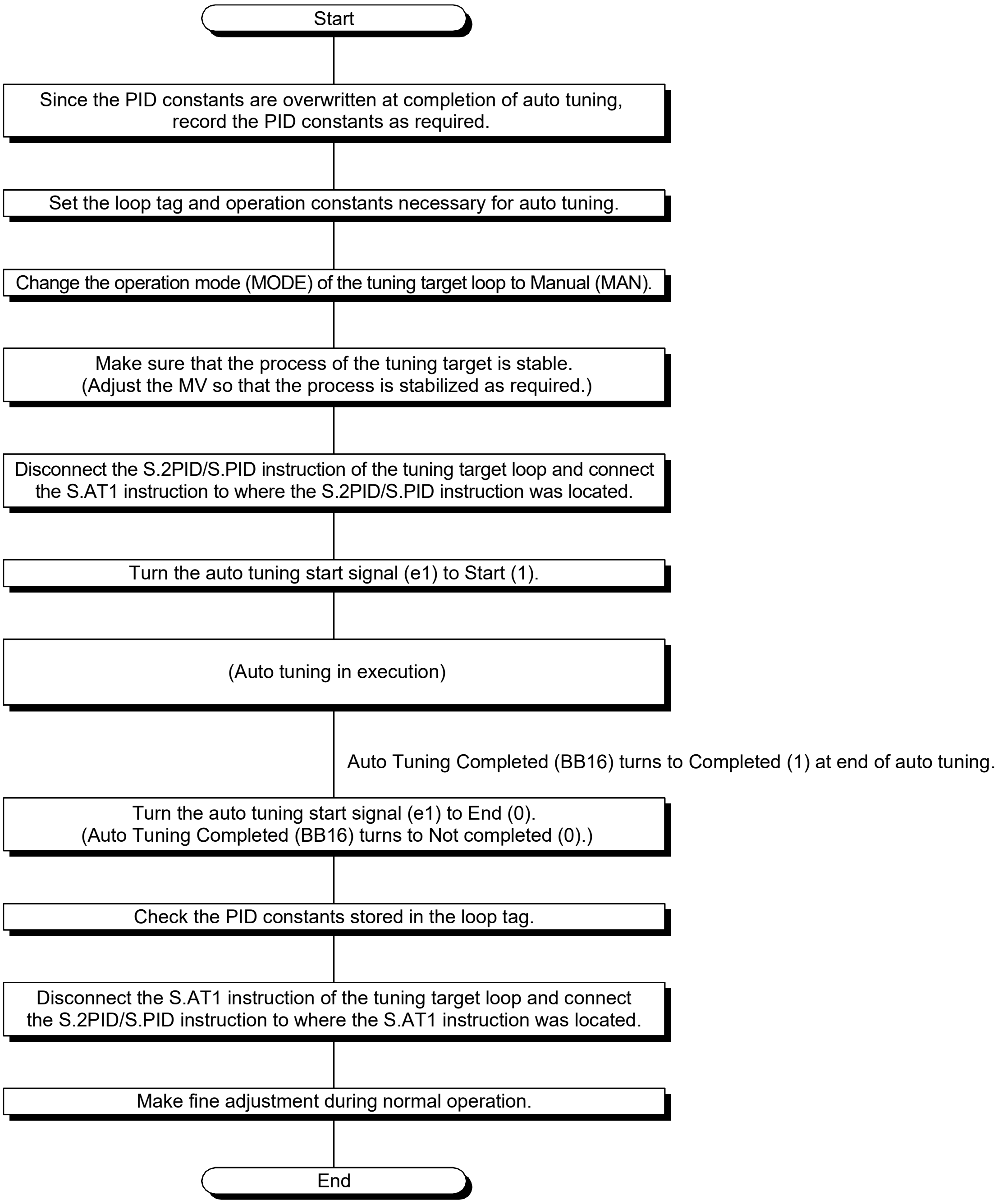
[Outline of stepped response process]

With no control operation being performed, change the manipulated value (MV) step by step and look how the process value (PV) varies.

1) When the manipulated value (MV) is changed step by step, the process value (PV) begins to vary slowly. Soon, the PV will vary faster, then vary slowly again, and finally settle at a fixed value.
2) Draw a tangent line at the place where the process value (PV) varies fastest, and find the points of intersection A, B where this tangent line crosses the horizontal axis corresponding to the first process value (θ_0) and last process value (θ_1).
   This provides the equivalent dead time (L) and equivalent time constant (T) as shown below.
3) From the equivalent time constant (T) and maximum process value width (Y), calculate the maximum slope (response speed) R = Y/T.
   Apply the equivalent dead time (L) and maximum slope (R) to the Ziegler and Nichols' adjustment rule, and calculate the proportional gain Kp (P), integral constant T_I (I) and derivative constant T_D (D).



Auto tuning procedure



Start

Since the PID constants are overwritten at completion of auto tuning, record the PID constants as required.

Set the loop tag and operation constants necessary for auto tuning.

Change the operation mode (MODE) of the tuning target loop to Manual (MAN).

Make sure that the process of the tuning target is stable.
(Adjust the MV so that the process is stabilized as required.)

Disconnect the S.2PID/S.PID instruction of the tuning target loop and connect the S.AT1 instruction to where the S.2PID/S.PID instruction was located.

Turn the auto tuning start signal (e1) to Start (1).

(Auto tuning in execution)

Auto Tuning Completed (BB16) turns to Completed (1) at end of auto tuning.

Turn the auto tuning start signal (e1) to End (0).
(Auto Tuning Completed (BB16) turns to Not completed (0).)

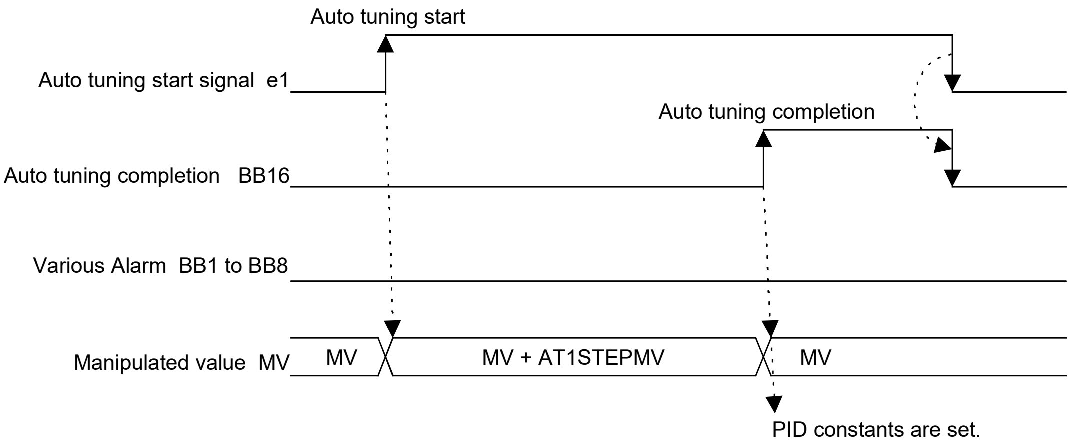
Check the PID constants stored in the loop tag.

Disconnect the S.AT1 instruction of the tuning target loop and connect the S.2PID/S.PID instruction to where the S.AT1 instruction was located.

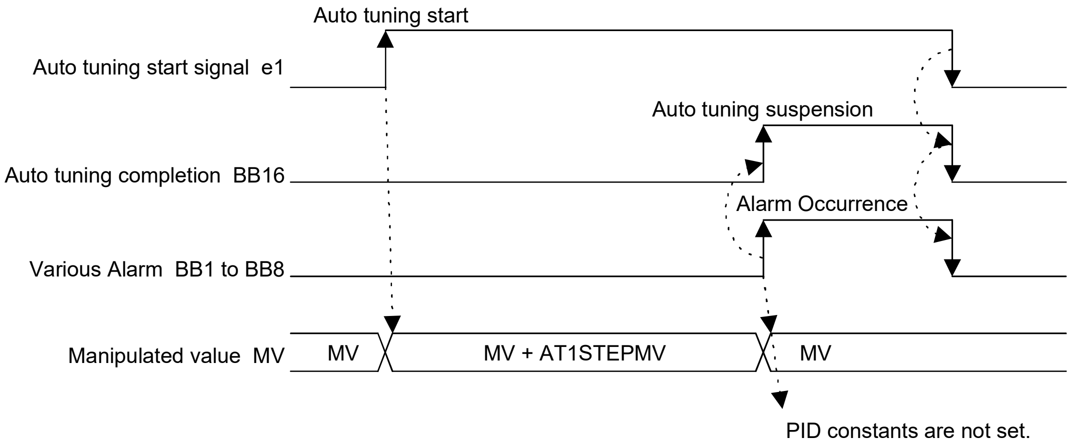
Make fine adjustment during normal operation.

End

(1) Time chart from auto tuning start until normal completion

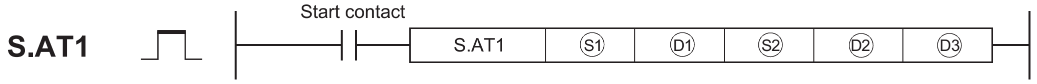


(2) Time chart from auto tuning start until stop due to alarm occurrence

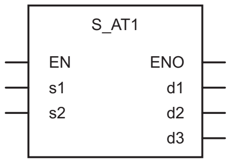


13.1 S. AT1

Ladder diagram



Structured ladder/FBD



Structured text language

ENO:=S_AT1(EN,s1,s2,d1,d2,d3);

Input argument	EN	: Execution condition	: Bit
	S1	: Input data start device	: Array of any 16-bit data (0..2)
	S2	: Operation constant start device	: Any 16-bit data
Output argument	ENO	: Execution result	: Bit
	D1	: Block memory start device	: Any 16-bit data
	D2	: Loop tag memory start device	: Array of any 16-bit data (0..127)
	D3	: Local work memory start device	: Array of any 16-bit data (0..21)

Setting data	Internal devices		R, ZR	J□\□		U□\G□	Zn	Constant	Other
	Bit	Word		Bit	Word				
S1	—	○		—					
D1	—	○		—					
S2	—	○		—					
D2	—	○		—					
D3	—	○		—					

Function

Performs auto tuning and makes the initial setting of the PID constants.

Set Data

(1) Data specified in S. AT1 instruction

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Input data	S1+0 +1	E1	Input value	-999999 to 999999	%	Real number	—	U
	+2	e1	Auto tuning start signal	b15 b12 b8 b4 b0 – – – – – – – – – – – – – – – e1 0: Stop/end 1: Start	—	BIN 16bit	0	U
Block memory	D1+0	BB		—				
		BB1	Alarm	b15 b12 b8 b4 b0 BB16 – – – – – – – BB8 BB7 BB6 BB5 BB4 BB3 BB2 BB1 BB16 (0: Incomplete) (1: Complete) BB1 to BB8 (0: Without alarm) (1: With alarm)	—	BIN 16bit	—	S
		BB2	Input upper limit alarm					
		BB3	Input lower limit alarm					
		BB4	Output upper limit alarm					
		BB5	Output lower limit alarm					
		BB6	Out time alarm					
		BB7	Operation mode alarm					
		BB8	Identification alarm					
		BB16	Auto tuning completion					
Operation constant	S2+0	PN	Operation mode	0: Reverse operation 1: Forward operation	—	BIN 16bit	0	U
Loop tag memory [*2]	D2+1	MODE	Operation mode	0 to FFFF$_H$ b15 b12 b8 b4 b0 – – – – – CSV CMV CCB CAB CMB CAS AUT MAN LCC LCA LCM	—	BIN 16bit	8$_H$	S/U
	+3	ALM	Alarm detection	b15 b12 b8 b4 b0 – SPA – – – – – HHA LLA PHA PLA – – – – – SPA 0: Loop RUN 1: Loop STOP HHA, LLA, PHA, PLA (0: Without alarm) (1: With alarm)	—	BIN 16bit	4000$_H$	S/U
	+12 +13	MV	Manipulated value	-10 to 110	%	Real number	0.0	S/U
	+18 +19	MH	Output upper limit value	-10 to 110	%	Real number	100.0	U
	+20 +21	ML	Output lower limit value	-10 to 110	%	Real number	0.0	U

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[*2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

Specified position		Symbol	Name	Recommended range [*1]	Unit	Data format	Standard value	Set by
Loop tag memory [*2]	(D2)+52 +53	P	Gain	0 to 999999	—	Real number	1.0	S/U
	+54 +55	I	Integral constant	0 to 999999	s	Real number	10.0	S/U
	+56 +57	D	Derivative constant	0 to 999999	s	Real number	0.0	S/U
	+70 +71	AT1 STEP MV	Step manipulated value for AT1	-100 to 100	%	Real number	0.0	U
	+72 +73	AT1ST	Sampling cycle for AT1	0 to 999999 Note that $\frac{\text{AT1ST}}{\Delta T} \leqq 32767$	s	Real number	1.0	U
	+74 +75	AT1 TOUT1	Time-out time for AT1	0 to 999999 Note that $\frac{\text{AT1TOUT1}}{\Delta T} \leqq 32767$	s	Real number	100.0	U
	+76 +77	AT1 TOUT2	After maximum slope time-out time for AT1	0 to 999999 Note that $\frac{\text{AT1TOUT2}}{\Delta T} \leqq 32767$	s	Real number	10.0	U
Local work memory [*3]	(D3)+0 ⋮ +21	—	System area	Used by the system as a work area.	—	—	—	S

[*1] The data of the item(s) where the values within the recommended range are given in the parentheses are stored by the system. Users cannot set the data.

[*2] The loop tag memory and loop tag past value memory occupy a total of 128 words. (Refer to Page 27, Section 3.3.1 for details.)

[*3] The applications of the loop tag past value memory are indicated below.

Specified position	contents
(S3)+0	Sampling cycle counter initial preset flag
+1	Sampling cycle counter[*4]
+2	Time-out time counter initial preset flag
+3	Time-out time counter[*4]
+4	After maximum slope time-out time counter initial preset flag
+5	After maximum slope time-out time counter[*4]
+6	Step manipulated value preset flag
+7	Counter from auto tuning start
+8 +9	Auto tuning start-time PV0
+10 +11	PV_{n-1} (Last process value)
+12 +13	Maximum slope value
+14 +15	Maximum slope-time counter
+16 +17	Maximum slope-time PV
+18 +19	R (Response speed)
+20 +21	L (Equivalent dead time)

When control is to be started from the initial status, the data must be cleared with the sequence program.

[*4] The sampling cycle counter, time-out time counter, and after maximum slope time-out time counter round off the data to the nearest whole number.

(2) Execution cycle (ΔT)

Set the execution cycle in SD1500 and SD1501 as a real number.

Processing contents

(1) Start signal judgement processing

Any of the following processings is performed depending on the statuses of the auto tuning start signal (e1) and auto tuning completed (BB16).

e1	BB16	Processing
0	0	BB1 to BB8 of BB are turned to 0. When the stepped manipulated value preset flag is 1, the following processing is performed. MV = MV - AT1STEPMV The S. AT1 instruction is terminated.
1	0	"(2) Loop stop processing" is performed.
0	1	BB16 of BB is turned to 0. The S. AT1 instruction is terminated.
1	1	The S. AT1 instruction is terminated.

(2) Loop stop processing

(a) Setting 1 in SPA of the alarm detection (ALM) selects a loop stop.
A loop stop performs the following processing and terminates the S. AT1 instruction.

1) The auto tuning completed (BB16) is turned to 1.

2) When the stepped manipulated value preset flag is 1, the following processing is performed.
MV = MV - AT1STEPMV

(b) Setting 0 in SPA of the alarm detection (ALM) selects a loop run.
A loop run performs "(3) Mode judgement processing".

(3) Mode judgement processing

Either of the following processings is performed depending on the operation mode (MODE) setting.

(a) When the operation mode (MODE) is any of AUT, CAB, CAS, CCB, CSV, LCA and LCC, the following processing is performed and the S. AT1 instruction is terminated.

1) The operation mode alarm (BB7) is turned to 1.

2) The auto tuning completed (BB16) is turned to 1.

3) When the stepped manipulated value preset flag is 1, the following processing is performed.
MV = MV - AT1STEPMV

(b) When the operation mode (MODE) is any of MAN, CMB, CMV and LCM, "(4) Input check processing" is performed.

(4) Input check processing

Either of the following processings is performed depending on the alarm detection (ALM) setting.

(a) If either of PHA and HHA of the alarm detection (ALM) is 1, the following processing is performed and the S. AT1 is terminated.

1) The input upper limit alarm (BB3) is turned to 1.

2) The auto tuning completed (BB16) is turned to 1.

(b) If either of PLA and LLA of the alarm detection (ALM) is 1, the following processing is performed and the S. AT1 is terminated.

1) The input lower limit alarm (BB3) is turned to 1.

2) The auto tuning completed (BB16) is turned to 1.

(5) Time-out judgement processing

Whether the auto tuning processing has reached the AT1 time-out time (AT1TOUT1) or not is judged.

(a) If the AT1 time-out time (AT1TOUT1) is reached, the following processing is performed and the S. AT1 is terminated.

1) The time-out alarm (BB6) is turned to 1.

2) The auto tuning completed (BB16) is turned to 1.

(b) If the AT1 time-out time (AT1TOUT1) is not reached, "(6) After maximum slope time-out judgment processing" is performed.

(6) After maximum slope time-out judgment processing

Whether the auto tuning processing has reached the AT1 after maximum slope time-out time (AT1TOUT2) or not is judged.

However, if the after maximum slope time-out time counter initial preset flag is 0, the processing in (c) is performed.

(a) If the AT1 after maximum slope time-out time (AT1TOUT2) is reached, "(10) Identification processing" is performed.

(b) If the AT1 after maximum slope time-out time (AT1TOUT2) is not reached, "(7) Stepped manipulated value set processing" is performed.

(c) If the after maximum slope time-out time counter initial preset flag is 0, "(7) Stepped manipulated value set processing" is performed.

(7) Stepped manipulated value set processing

Whether the stepped manipulated value is "set (1)" or "not set (0)" is judged from the stepped manipulated value preset flag.

(a) If the stepped manipulated value preset flag is 0, the following processing is performed and the S. AT1 is terminated.

1) The AT1 stepped manipulated value (AT1STEPMV) is added to the manipulated value (MV).

T1 = MV + AT1STEPMV

In the upper/lower limiter, the following operation is performed and the result of the operation is output to BB4 and BB5.

Condition	Result				Processing after upper/lower limiter
	BB4	BB5	BB16	MV	
T1 > MH	1	0	1	Original MV unchanged	S. AT1 instruction is terminated.
T1 < ML	0	1	1	Original MV unchanged	
$ML \leq T1 \leq MH$	0	0	0	T1	The processing in 2) and later is performed.

2) The stepped manipulated value preset flag is turned to 1.

3) The counter from auto tuning start is cleared to 0.

4) The input value (E1) is stored into the auto tuning start-time PV0.

5) The input value (E1) is stored into the last process value (PV_{n-1}).

6) The maximum slope value, maximum slope-time counter, maximum slope-time PV, response speed (R) and equivalent dead time (L) are cleared to 0.

(b) If the stepped manipulated value preset flag is 1 "(8) Sampling cycle judgement processing" is performed.

(8) Sampling cycle judgment processing

Whether the sampling cycle is reached or not is judged from the AT1 sampling cycle (AT1ST).

(a) If the sampling cycle is not reached, the S. AT1 instruction is terminated.

(b) If the sampling cycle is reached, "(9) Response waveform observation processing" is performed.

(9) Response waveform observation processing

The following processing is performed for the input value (E1).

(a) Response waveform observation

1) The counter from auto tuning start is incremented.

2) The following processing is performed according to the input value (E1) and last process value (PV_{n-1}).

Reverse operation (PN = 0)	$T2 = E1 - PV_{n-1}$
Forward operation (PN = 1)	

3) The input value (E1) is stored into the last process value (PV_{n-1}).

(b) Maximum slope value

Depending on the slope (T2), the following processing is performed and the S. AT1 instruction is terminated.

1) If reverse operation is performed (PN = 0) and AT1 stepped manipulated value (AT1STEPMV) ≥ 0 or forward operation is performed (PN = 1) and AT1 stepped manipulated value (AT1STEPMV) < 0

Condition	Processing
Maximum slope value slope ≤ (T2)	• Maximum slope value = slope (T2) • Maximum slope-time counter = counter from auto tuning start • Maximum slope-time PV = input value (E1) • After maximum slope time-out count value is reset and count is restarted.
Maximum slope value > Slope(T2)	Maximum slope value remains unchanged from the last value.

2) If forward operation is performed (PN = 1) and AT1 stepped manipulated value (AT1STEPMV) ≥ 0 or reverse operation is performed (PN = 0) and AT1 stepped manipulated value (AT1STEPMV) = 0

Condition	Processing
Maximum slope value ≥ Slope(T2)	• Maximum slope value = slope (T2) • Maximum slope-time counter = counter from auto tuning start • Maximum slope-time PV = input value (E1) • After maximum slope time-out count value is reset and count is restarted.
Maximum slope value < Slope(T2)	Maximum slope value remains unchanged from the last value.

(10) Identification processing

Using the maximum slope value, the following processing is performed.

(a) Response speed

1) The response speed for calculation (R') and response speed (R) are calculated with the following expression.

$$R' = \frac{\begin{array}{c}\text{Maximum}\\ \text{slope value(\%)}\end{array}}{\text{AT1ST(s)}}, \quad R = \frac{|R'|}{100}\text{(/s)}$$

2) If R = 0, the following processing is performed and the S. AT1 instruction is terminated.

The identification alarm (BB8) is turned to 1.

The auto tuning completed (BB16) is turned to 1.

When the stepped manipulated value preset flag is 1, the following processing is performed.

MV = MV - AT1STEPMV

(b) Equivalent dead time

1) The segment (b) made by the Y axis and the equivalent dead time (L) provided when the tangent line is drawn at the response speed for calculation (R') are calculated with the following expression.

b = (maximum slope-time PV) - R' × (maximum slope counter) × AT1ST

$$L = \frac{\text{(Auto tuning start-time PV0)-b}}{R'}$$

2) If L ≤ 0, the following processing is performed and the S. AT1 instruction is terminated.

The identification alarm (BB8) is turned to 1.

The auto tuning completed (BB16) is turned to 1.

When the stepped manipulated value preset flag is 1, the following processing is performed.

MV = MV - AT1STEPMV

(11) PID constant calculation processing

The response speed (R), equivalent dead time (L) and AT1 stepped manipulated value (AT1STEPMV) are assigned to the adjustment rule to calculate the PID constants.

(a) Control system

The control system is selected according to the integral constant T_I (I) and derivative constant T_D (D).

Integral constant T_I (I)	Derivative constant T_D (D)	Control method
$T_I \leq 0$	—	Proportional control (P operation) only
$T_I > 0$	$T_D \leq 0$	PI control (PI operation)
	$T_D > 0$	PID control (PID operation)

(b) Adjustment rule

The ZN process: adjustment rule based on the stepped response of Ziegler and Nichols is used.

Control method	Rate example gain Kp (P)	Integral constant T_I (I)	Derivative constant T_D (D)
P	$\frac{1}{R \times L} \times \frac{\lvert AT1STEPMV \rvert}{100}$	0	0
PI	$\frac{0.9}{R \times L} \times \frac{\lvert AT1STEPMV \rvert}{100}$	3.33 × L	0
PID	$\frac{1.2}{R \times L} \times \frac{\lvert AT1STEPMV \rvert}{100}$	2 × L	0.5 × L

(c) PID constant storage

The following processing is performed and the S. AT1 instruction is terminated.

1) The PID constants are stored into the gain (P), integral constant (I) and derivative constant (D).

2) The auto tuning completed (BB16) is turned to 1.

3) The AT1 stepped manipulated value (AT1STEPMV) is subtracted from the manipulated value (MV), and the result is stored into the manipulated value (MV).
MV = MV - AT1STEPMV

Operation Error

In the following cases, the error flag (SM0) turns ON and the error code is stored in SD0.

Error code	Error definition	QnPHCPU	QnPRHCPU	QnUDPVCPU
4100	When an operation error occurs	○	○	—
	When the value of (S1) is either a non-numeric or non-normalized number	○	○	—
	When AT1ST < 0, AT1TOUT1 < 0 or AT1TOUT2 < 0	○	○	○
	When the execution cycle (SD1500) < 0	○	○	○
4140	When the value of (S1) is either a non-numeric or non-normalized number	—	—	○
4141	When an operation error occurs	—	—	○

CHAPTER 14 ERROR CODES

This chapter describes the definitions of the errors that will occur in the CPU module and the compensation operation to be taken for the errors.

14.1 List of Error Codes

There is the following process control instruction error.

Definition	Applicable CPU module	Error code
When the operation target data is out of the recommended range	QnPHCPU/QnPRHCPU/QnUDPVCPU	4100
When the operation target data is a non-numeric or non-normalized number	QnPHCPU/QnPRHCPU	4100
	QnUDPVCPU	4140
When an error occurs midway through operation	QnPHCPU/QnPRHCPU	4100
	QnUDPVCPU	4141

When the above error occurs, the following information is stored in SD1502 and SD1503.

- SD1502: Detailed error code that occurred in a process control instruction
- SD1503: Processing number of the instruction that an error occurred

Codes of errors that occur in process control instructions (The corresponding error code is stored in SD1502.)

Error code	Error definition	Detailed error code stored in SD1502	Cause	Processing
4100	There is either a non-numeric or non-normalized number.	1	Set data, such as operation constant, loop tag memory, loop tag past value memory or execution cycle, has a problem.	Check/correct the set data.
	Sign error (The number is negative.)	2		
	Numerical value error (The value is out of the range.)	3		
	The value is not an integer.	4		
	Tried to divide by 0.	5		
	An overflow occurred.	6		

Processing numbers of each instruction (The number is stored in SD1503.)

Instruction	Processing numbers stored in SD1503							
	1	2	3	4	5	6	7	8
S. IN	Range check	Input limiter	Engineering value reverse conversion	Digital Filter				
S. OUT1	Input addition processing	Change rate, upper/lower limiter	Reset windup	Output conversion				
S. OUT2		Change rate, upper/lower limiter		Output conversion				
S. DUTY	Input addition processing	Change rate, upper/lower limiter	Reset windup	Output ON time conversion				
S. BC		Change rate check						
S. PSUM		Integration value operation	Output conversion					
S. PID	Control cycle judgment	SV setting processing	Tracking processing	Gain (Kp) operation	PID operation	Deviation check		
S. 2PID	Control cycle judgment	SV setting processing	Tracking processing	Gain (Kp) operation	PID operation 1) *1	PID operation 2) *2	PID operation 3) *3	Deviation check
S. PIDP	Control cycle judgment	SV setting processing	Tracking processing	Gain (Kp) operation	PIDP operation	Deviation check	Change rate, upper/ lower limiter	Output conversion
S. SPI	Operation time monitor	SV setting processing	Tracking processing	Gain (Kp) operation	SPI operation	Deviation check		
S. IPD	Control cycle judgment	SV setting processing	Tracking processing	Gain (Kp) operation	IPD operation	Deviation check		
S. BPI	Control cycle judgment	SV setting processing	Tracking processing	Gain (Kp) operation	BPI operation	Deviation check		
S. R	Control cycle judgment	Engineering value conversion	Tracking processing	Change rate limiter	Ratio operation			
S. PHPL	Engineering value reverse conversion	Upper/lower limit check	Change rate check	Engineering value conversion	Loop stop			
S. ONF2	Control cycle judgment	SV setting processing	Tracking processing	MV compensa-tion		2-position ON/ OFF control		
S. ONF3	Control cycle judgment	SV setting processing	Tracking processing	MV compensa-tion		3-position ON/ OFF control		
S. PGS	Operation constant check	SV count up	MVPGS operation	Output processing				
S. SEL	Engineering value conversion		Engineering value reverse conversion	Change rate, upper/ lower limiter	Output conversion			
S. AT1	Input check	Time out judgment		Step manipulated value set	Sampling cycle judgment	Response waveform observation	Identification processing	PID constant calculation

*1 Indicates the operation processing of Bn or Cn.

*2 Indicates the operation processing of Dn.

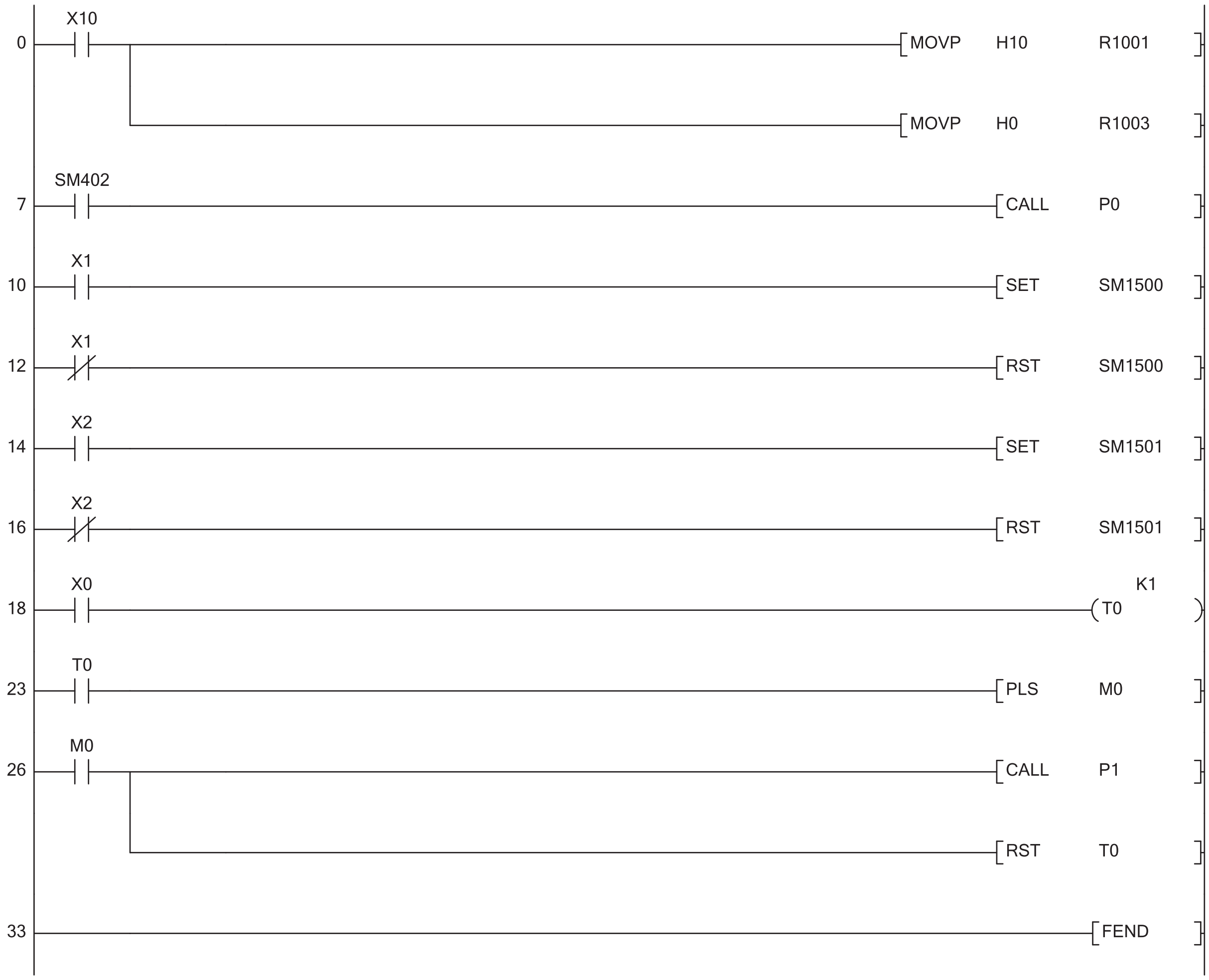
*3 Indicates the operation processing of ΔMV.

Processing No. 1 is stored if an error occurs in the instruction that is not indicated in the above table.

APPENDICES

Appendix 1 Program Example

In the following program, the operation mode at power-on is in manual mode. When X10 turns ON, the mode switches to automatic mode and the module starts PID control.



(0) Set the operation mode to AUTO.
Set the alarm detection to 0.

(7) Jump to the P0 label when SM402 is on.

(10) Set the last value hold processing.

(12) Reset the last value hold processing.

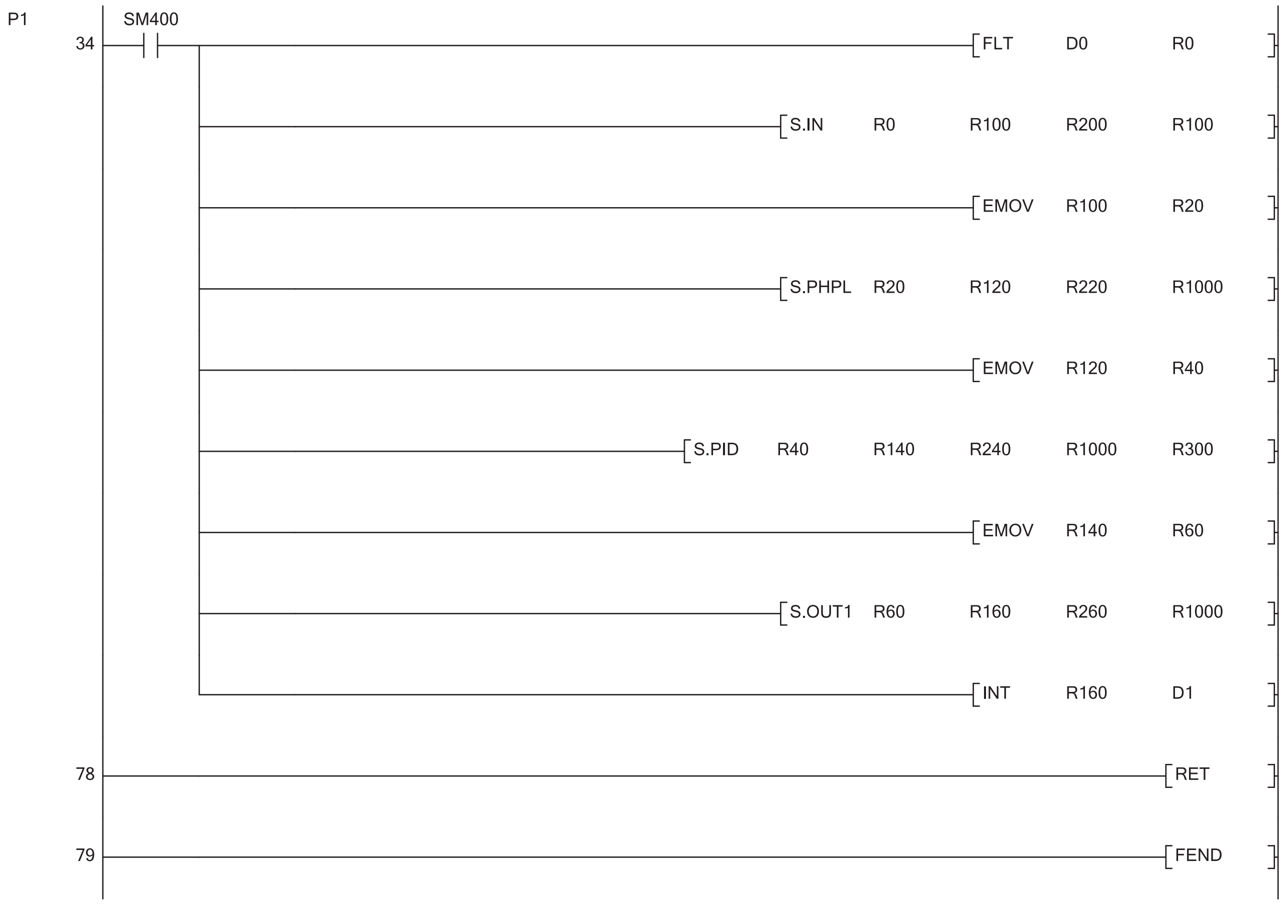
(14) Set the output value hold processing.

(16) Reset the output value hold processing.

(18) Adjust to the time set for the execution cycle.

(26) Jump to the P1 label when M0 is on.
Turning M0 on clears T0 to 0.

(33) The end of the sequence program in which SM402 is off is indicated.

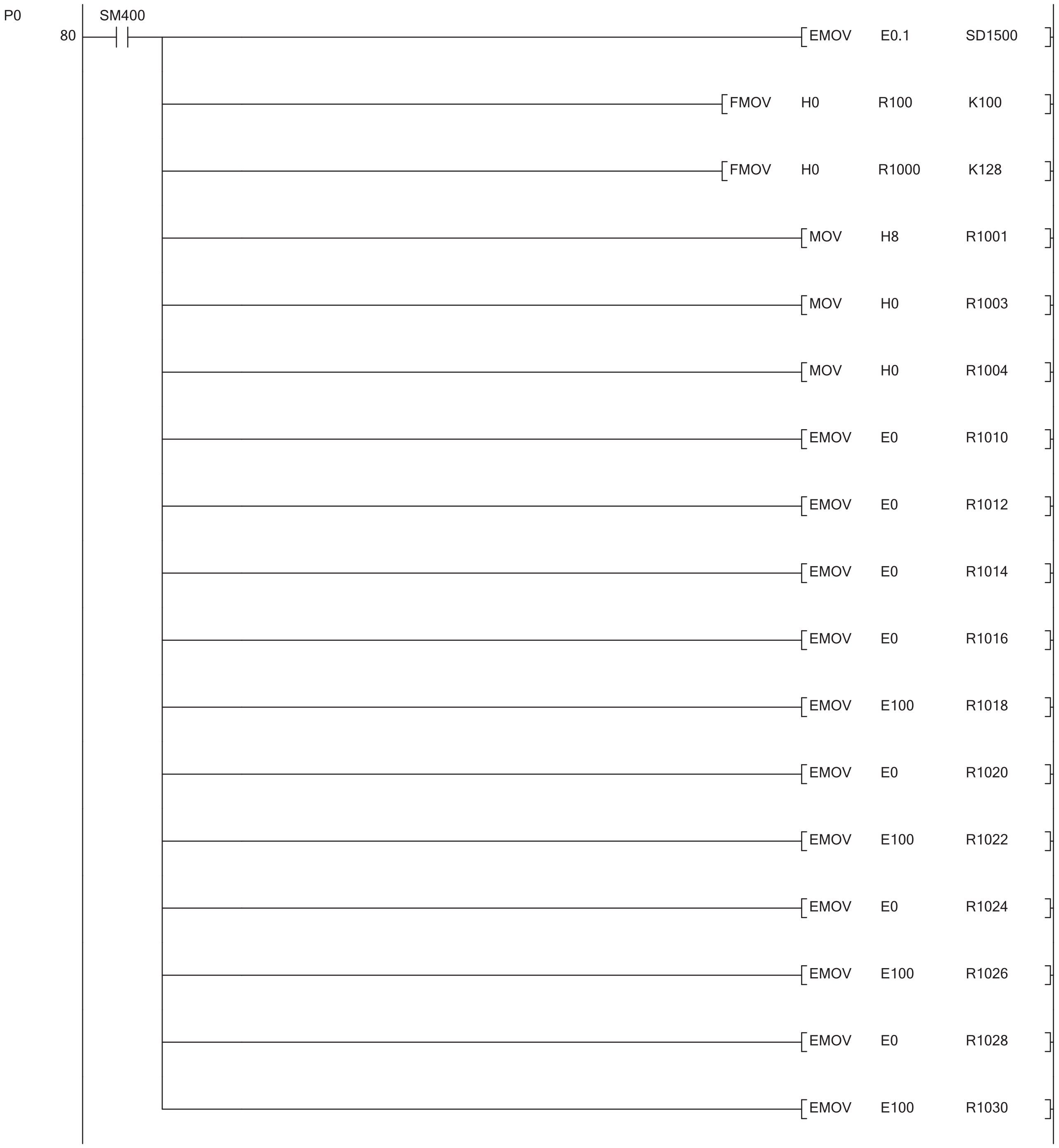


```
P1      SM400
34 ──┤ ├──┬────────────────────────────[FLT     D0      R0                ]
          ├───────────────────[S.IN     R0      R100    R200    R100      ]
          ├────────────────────────────[EMOV    R100    R20               ]
          ├───────────────────[S.PHPL   R20     R120    R220    R1000     ]
          ├────────────────────────────[EMOV    R120    R40               ]
          ├──────────[S.PID    R40      R140    R240    R1000   R300      ]
          ├────────────────────────────[EMOV    R140    R60               ]
          ├───────────────────[S.OUT1   R60     R160    R260    R1000     ]
          └────────────────────────────[INT     R160    D1                ]
78 ────────────────────────────────────────────────────[RET               ]
79 ────────────────────────────────────────────────────[FEND              ]
```

(34) Change the D0 value to a real number and store 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.

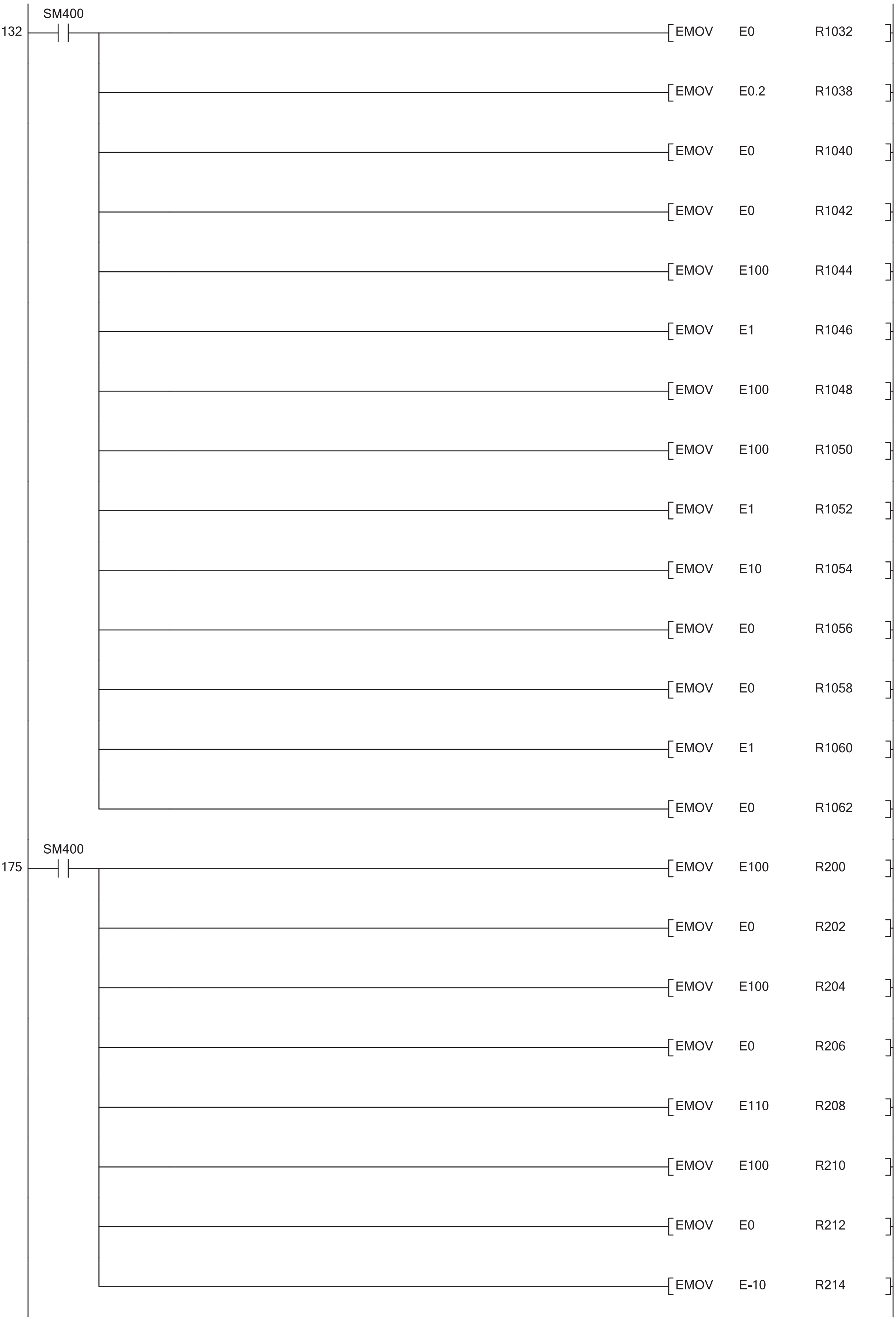
(78) The subroutine program ends.

(79) The main routine program ends.



```
P0      SM400
80  ──┤ ├──┬──────────────────[EMOV   E0.1   SD1500 ]
           ├─────────────[FMOV   H0     R100    K100 ]
           ├─────────────[FMOV   H0     R1000   K128 ]
           ├──────────────────[MOV    H8     R1001  ]
           ├──────────────────[MOV    H0     R1003  ]
           ├──────────────────[MOV    H0     R1004  ]
           ├──────────────────[EMOV   E0     R1010  ]
           ├──────────────────[EMOV   E0     R1012  ]
           ├──────────────────[EMOV   E0     R1014  ]
           ├──────────────────[EMOV   E0     R1016  ]
           ├──────────────────[EMOV   E100   R1018  ]
           ├──────────────────[EMOV   E0     R1020  ]
           ├──────────────────[EMOV   E100   R1022  ]
           ├──────────────────[EMOV   E0     R1024  ]
           ├──────────────────[EMOV   E100   R1026  ]
           ├──────────────────[EMOV   E0     R1028  ]
           └──────────────────[EMOV   E100   R1030  ]
```

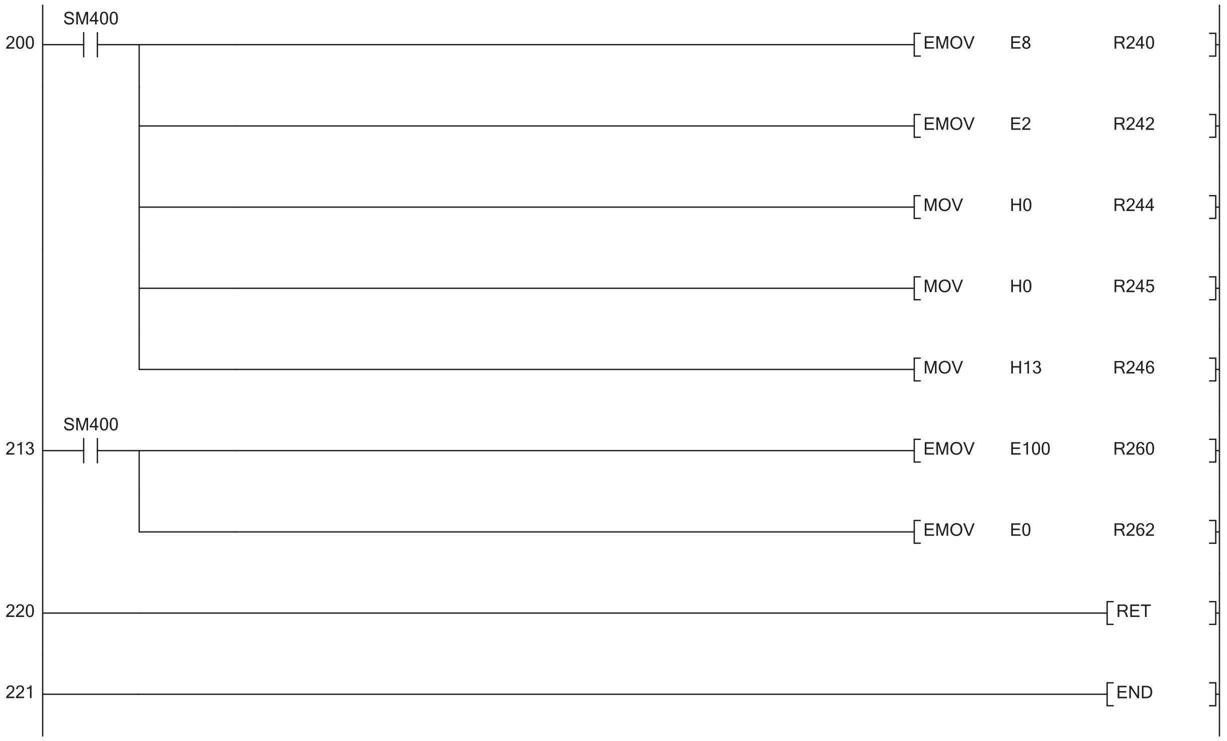
(80) Set the execution cycle to 0.1 seconds.
Clear the output values in S.IN, S.PHPL, S.PID, and S.OUT1 to 0.
Clear the loop tag to 0.
Set the default value of the loop tag.



```
     SM400
132 --| |--+--[EMOV  E0    R1032 ]
           +--[EMOV  E0.2  R1038 ]
           +--[EMOV  E0    R1040 ]
           +--[EMOV  E0    R1042 ]
           +--[EMOV  E100  R1044 ]
           +--[EMOV  E1    R1046 ]
           +--[EMOV  E100  R1048 ]
           +--[EMOV  E100  R1050 ]
           +--[EMOV  E1    R1052 ]
           +--[EMOV  E10   R1054 ]
           +--[EMOV  E0    R1056 ]
           +--[EMOV  E0    R1058 ]
           +--[EMOV  E1    R1060 ]
           +--[EMOV  E0    R1062 ]
     SM400
175 --| |--+--[EMOV  E100  R200  ]
           +--[EMOV  E0    R202  ]
           +--[EMOV  E100  R204  ]
           +--[EMOV  E0    R206  ]
           +--[EMOV  E110  R208  ]
           +--[EMOV  E100  R210  ]
           +--[EMOV  E0    R212  ]
           +--[EMOV  E-10  R214  ]
```

(132) Set the default value of the loop tag.

(175) Set the operation constant of the S.IN instruction.



(200) Set the operation constant of the S.PID instruction.
(213) Set the operation constant of the S.OUT1 instruction.
(220) The subroutine program ends.

Appendix 2 Loop Tag Memory List

Appendix 2.1 PID control (SPID), 2-degree-of-freedom PID control (S2PID), sample PI control (SSPI)

Instruction used	Offset	Item	Name	Recommended range	Unit	Data storage	
						SPID S2PID	SSPI
	+0	—	—	—	—	—	—
	1	MODE[*1]	Operation mode	0 to FFFF$_H$	—	S/U	S/U
	2	—	—	—	—	—	—
	3	ALM[*1]	Alarm detection	0 to FFFF$_H$	—	S/U	S/U
	4	INH[*1]	Alarm detection inhibition	0 to FFFF$_H$	—	S/U	S/U
	5	—	—	—	—	—	—
	6	—	—	—	—	—	—
	7	—	—	—	—	—	—
	8	—	—	—	—	—	—
	9	—	—	—	—	—	—
S. PHPL	10 11	PV	Process value	(RL to RH)	—	S	S
S. OUT1/S. DUTY	12 13	MV	Manipulated value	-10 to 110	%	S/U	S/U
S. PID/S. 2PID/S. SPI	14 15	SV	Set value	RL to RH	—	U	U
S. PID/S. 2PID/S. SPI	16 17	DV	Deviation	(-110 to 110)	%	S	S
S. OUT1/S. DUTY	18 19	MH	Output upper limit value	-10 to 110	%	U	U
S. OUT1/S. DUTY	20 21	ML	Output lower limit value	-10 to 110	%	U	U
S. PHPL/S. PID/S. 2PID/S. SPI	22 23	RH	Engineering value upper limit	-999999 to 999999	—	U	U
S. PHPL/S. PID/S. 2PID/S. SPI	24 25	RL	Engineering value lower limit	-999999 to 999999	—	U	U
S. PHPL	26 27	PH	Upper limit alarm set value	RL to RH PL < PH	—	U	U
S. PHPL	28 29	PL	Lower limit alarm value	RL to RH PL < PH	—	U	U
S. PHPL	30 31	HH	Upper upper limit alarm value	RL to RH PH ≤ HH	—	U	U
S. PHPL	32 33	LL	Lower lower limit alarm value	RL to RH LL ≤ PL	—	U	U
	34 35	—	—	—	—	—	—
	36 37	—	—	—	—	—	—
S. IN	38 39	α	Filter coefficient	0 to 1	—	U	U
S. PHPL	40 41	HS	Upper/lower limit alarm hysteresis	0 to 999999	%	U	U
S. PHPL	42 43	CTIM	Change rate alarm check time	0 to 999999	s	U	U
S. PHPL	44 45	DPL	Change rate alarm value	0 to 100	%	U	U
S. PID/S. 2PID/S. SPI	46 47	CT/ST	Control cycle/Operation time	0 to 999999	s	U (Set CT)	U (Set ST)

Instruction used	Offset	Item	Name	Recommended range	Unit	Data storage	
						SPID S2PID	SSPI
S. OUT1/S.DUTY	48 49	DML	Output change rate limit value	0 to 100	%	U	U
S. PID/S. 2PID/ S. SPI	50 51	DVL	Deviation limit value	0 to 100	%	U	U
S. PID/S. 2PID/ S. SPI	52 53	P	Gain	0 to 999999	—	U	U
S. PID/S. 2PID/ S. SPI/S. OUT1/ S. DUTY	54 55	I*2	Integral constant	0 to 999999	s	U	U
S. PID/S. 2PID/ S. SPI	56 57	D/STHT	Derivative constant/ Sample cycle	0 to 999999	s	U (D Setting)	U (STHT Setting)
S. PID/S. 2PID/ S. SPI	58 59	GW	Gap width	0 to 100	%	U	U
S. PID/S. 2PID/ S. SPI	60 61	GG	Gap gain	0 to 999999	—	U	U
S. PID/S. 2PID/ S. SPI/S. OUT1/S. DUTY	62 63	MVP	MV inside operation value	(-999999 to 999999)	%	S	S
S. 2PID	64 65	α	2Degree-of-freedom parameter α	0 to 1	—	U	—
S. 2PID	66 67	β	2Degree-of-freedom parameter β	0 to 1	—	U	—
S. DUTY	68 69	CTDUTY	Control output cycle	0 to 999999	s	U	—

*1 MODE, ALM, and INH are shared among the instructions.

*2 The following instructions share 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

Appendix 2.2 I-PD control (SIPD), blend PI control (SBPI)

Instruction used	Offset	Item	Name	Recommended range	Unit	Data storage	
						SIPD	SBPI
	+0	—	—	—	—	—	—
	1	MODE*1	Operation mode	0 to FFFF$_H$	—	S/U	S/U
	2	—	—	—	—	—	—
	3	ALM*1	Alarm detection	0 to FFFF$_H$	—	S/U	S/U
	4	INH*1	Alarm detection inhibition	0 to FFFF$_H$	—	S/U	S/U
	5	—	—	—	—	—	—
	6	—	—	—	—	—	—
	7	—	—	—	—	—	—
	8	—	—	—	—	—	—
	9	—	—	—	—	—	—
S. PHPL	10 11	PV	Process value	(RL to RH)	—	S	S
S. OUT1	12 13	MV	Manipulated value	-10 to 110	%	S/U	S/U
S. IPD/S. BPI	14 15	SV	Set value	RL to RH	—	U	U
S. IPD/S. BPI	16 17	DV	Deviation	(-110 to 110)	%	S	S
S. OUT1	18 19	MH	Output upper limit value	-10 to 110	%	U	U
S. OUT1	20 21	ML	Output lower limit value	-10 to 110	%	U	U
S. PHPL/S. IPD/S. BPI	22 23	RH	Engineering value upper limit	-999999 to 999999	—	U	U
S. PHPL/S. IPD/S. BPI	24 25	RL	Engineering value lower limit	-999999 to 999999	—	U	U
S. PHPL	26 27	PH	Upper limit alarm set value	RL to RH PL < PH	—	U	U
S. PHPL	28 29	PL	Lower limit alarm value	RL to RH PL < PH	—	U	U
S. PHPL	30 31	HH	Upper upper limit alarm value	RL to RH PH ≤ HH	—	U	U
S. PHPL	32 33	LL	Lower lower limit alarm value	RL to RH LL ≤ PL	—	U	U
	34 35	—	—	—	—	—	—
	36 37	—	—	—	—	—	—
S. IN	38 39	α	Filter coefficient	0 to 1	—	U	U
S. PHPL	40 41	HS	Upper/lower limit alarm hysteresis	0 to 999999	%	U	U
S. PHPL	42 43	CTIM	Change rate alarm check time	0 to 999999	s	U	U
S. PHPL	44 45	DPL	Change rate alarm value	0 to 100	%	U	U
S. IPD/S. BPI	46 47	CT	Control cycle	0 to 999999	s	U	U

Instruction used	Offset	Item	Name	Recommended range	Unit	Data storage	
						SIPD	SBPI
S. OUT1	48 49	DML	Output change rate limit value	0 to 100	%	U	U
S. IPD/S. BPI	50 51	DVL	Deviation limit value	0 to 100	%	U	U
S. IPD/S. BPI	52 53	P	Gain	0 to 999999	—	U	U
S. IPD/S. BPI S. OUT1	54 55	I[*2]	Integral constant	0 to 999999	s	U	U
S. IPD/S. BPI	56	D/SDV	Derivative constant/ DV cumulative value	D:0 to 999999	s	U	—
	57			SDV:-999999 to 999999	%	—	S
S. IPD/S. BPI	58 59	GW	Gap width	0 to 100	%	U	U
S. IPD/S. BPI	60 61	GG	Gap gain	0 to 999999	—	U	U
S. IPD/S. OUT1	62 63	MVP	MV inside operation value	(-999999 to 999999)	%	S	—

[*1] MODE, ALM, and INH are shared among the instructions.

[*2] The following instructions share the same value in I.

- S. IPD instruction and S. OUT1 instruction
- S. BPI instruction and S. OUT1 instruction

Appendix 2.3 Manual output (SMOUT), monitor (SMON)

Instruction used	Offset	Item	Name	Setting/Store range	Unit	Data storage	
						SMOUT	SMON
	+0	—	—	—	—	—	—
	1	MODE[*1]	Operation mode	0 to FFFF$_H$	—	S/U	S/U
	2	—	—	—	—	—	—
	3	ALM[*1]	Alarm detection	0 to FFFF$_H$	—	S/U	S/U
	4	INH[*1]	Alarm detection inhibition	0 to FFFF$_H$	—	—	S/U
	5	—	—	—	—	—	—
	6	—	—	—	—	—	—
	7	—	—	—	—	—	—
	8	—	—	—	—	—	—
	9	—	—	—	—	—	—
S. PHPL	10	PV	Process value	(RL to RH)	—	—	S
	11						
S. MOUT	12	MV	Manipulated value	-10 to 110	%	U	—
	13						
	14	—	—	—	—	—	—
	15						
	16	—	—	—	—	—	—
	17						
	18	—	—	—	—	—	—
	19						
	20	—	—	—	—	—	—
	21						
S. PHPL	22	RH	Engineering value upper limit	-999999 to 999999	—	—	U
	23						
S. PHPL	24	RL	Engineering value lower limit	-999999 to 999999	—	—	U
	25						
S. PHPL	26	PH	Upper limit alarm set value	RL to RH PL < PH	—	—	U
	27						
S. PHPL	28	PL	Lower limit alarm value	RL to RH PL < PH	—	—	U
	29						
S. PHPL	30	HH	Upper upper limit alarm value	RL to RH PH ≤ HH	—	—	U
	31						
S. PHPL	32	LL	Lower lower limit alarm value	RL to RH LL ≤ PL	—	—	U
	33						
	34	—	—	—	—	—	—
	35						
	36	—	—	—	—	—	—
	37						
S. IN	38	α	Filter coefficient	0 to 1	—	—	U
	39						
S. PHPL	40	HS	Upper/lower limit alarm hysteresis	0 to 999999	%	—	U
	41						
S. PHPL	42	CTIM	Change rate alarm check time	0 to 999999	s	—	U
	43						
S. PHPL	44	DPL	Change rate alarm value	0 to 100	%	—	U
	45						
	46	—	—	—	—	—	—
	47						

[*1] MODE, ALM, and INH are shared among the instructions.

Appendix 2.4 Manual output with monitor (SMWM), PIDP control (SPIDP)

Instruction used	Offset	Item	Name	Recommended range	Unit	Data storage	
						SMWM	SPIDP
	+0	—	—	—	—	—	—
	1	MODE[*1]	Operation mode	0 to FFFF_H	—	S/U	S/U
	2	—	—	—	—	—	—
	3	ALM[*1]	Alarm detection	0 to FFFF_H	—	S/U	S/U
	4	INH[*1]	Alarm detection inhibition	0 to FFFF_H	—	S/U	S/U
	5	—	—	—	—	—	—
	6	—	—	—	—	—	—
	7	—	—	—	—	—	—
	8	—	—	—	—	—	—
	9	—	—	—	—	—	—
S. PHPL	10 11	PV	Process value	(RL to RH)	—	S	S
S. MOUT/S. PIDP	12 13	MV	Manipulated value	-10 to 110	%	U	S/U
S. PIDP	14 15	SV	Set value	RL to RH	—	—	U
S. PIDP	16 17	DV	Deviation	(-110 to 110)	%	—	S
S. PIDP	18 19	MH	Output upper limit value	-10 to 110	%	—	U
S. PIDP	20 21	ML	Output lower limit value	-10 to 110	%	—	U
S. PHPL/S. PIDP	22 23	RH	Engineering value upper limit	-999999 to 999999	—	U	U
S. PHPL/S. PIDP	24 25	RL	Engineering value lower limit	-999999 to 999999	—	U	U
S. PHPL	26 27	PH	Upper limit alarm set value	RL to RH PL < PH	—	U	U
S. PHPL	28 29	PL	Lower limit alarm value	RL to RH PL < PH	—	U	U
S. PHPL	30 31	HH	Upper upper limit alarm value	RL to RH PH ≤ HH	—	U	U
S. PHPL	32 33	LL	Lower lower limit alarm value	RL to RH LL ≤ PL	—	U	U
	34 35	—	—	—	—	—	—
	36 37	—	—	—	—	—	—
S. IN	38 39	α	Filter coefficient	0 to 1	—	U	U
S. PHPL	40 41	HS	Upper/lower limit alarm hysteresis	0 to 999999	%	U	U
S. PHPL	42 43	CTIM	Change rate alarm check time	0 to 999999	s	U	U
S. PHPL	44 45	DPL	Change rate alarm value	0 to 100	%	U	U
S. PIDP	46 47	CT	Control cycle	0 to 999999	s	—	U

Instruction used	Offset	Item	Name	Recommended range	Unit	Data storage	
						SMWM	SPIDP
S. PIDP	48 49	DML	Output change rate limit value	0 to 100	%	—	U
S. PIDP	50 51	DVL	Deviation limit value	0 to 100	%	—	U
S. PIDP	52 53	P	Gain	0 to 999999	—	—	U
S. PIDP	54 55	I	Integral constant	0 to 999999	s	—	U
S. PIDP	56 57	D	Derivative constant	0 to 999999	s	—	U
S. PIDP	58 59	GW	Gap width	0 to 100	%	—	U
S. PIDP	60 61	GG	Gap gain	0 to 999999	—	—	U

*1 MODE, ALM, and INH are shared among the instructions.

Appendix 2.5 2 position ON/OFF control (SONF2), 3 position ON/OFF control (SONF3)

Instruction used	Offset	Item	Name	Recommended range	Unit	Data storage	
						SONF2	SONF3
	+0	—	—	—	—	—	—
	1	MODE[*1]	Operation mode	0 to FFFF_H	—	S/U	S/U
	2	—	—	—	—	—	—
	3	ALM[*1]	Alarm detection	0 to FFFF_H	—	S/U	S/U
	4	INH[*1]	Alarm detection inhibition	0 to FFFF_H	—	S/U	S/U
	5	—	—	—	—	—	—
	6	—	—	—	—	—	—
	7	—	—	—	—	—	—
	8	—	—	—	—	—	—
	9	—	—	—	—	—	—
S.PHPL	10 11	PV	Process value	(RL to RH)	—	S	S
S.ONF2/S.ONF3	12 13	MV	Manipulated value	-10 to 110	%	S/U	S/U
S.ONF2/S.ONF3	14 15	SV	Set value	RL to RH	—	U	U
S.ONF2/S.ONF3	16 17	DV	Deviation	(-110 to 110)	%	S	S
S.ONF2/S.ONF3	18 19	HSO	Hysteresis	0 to 999999	—	U	U
S.ONF3	20 21	HS1	Hysteresis	0 to 999999	—	—	U
S.PHPL	22 23	RH	Engineering value upper limit	-999999 to 999999	—	U	U
S.PHPL	24 25	RL	Engineering value lower limit	-999999 to 999999	—	U	U
S.PHPL	26 27	PH	Upper limit alarm set value	RL to RH PL < PH	—	U	U
S.PHPL	28 29	PL	Lower limit alarm value	RL to RH PL < PH	—	U	U
S.PHPL	30 31	HH	Upper upper limit alarm value	RL to RH PH ≤ HH	—	U	U
S.PHPL	32 33	LL	Lower lower limit alarm value	RL to RH LL ≤ PL	—	U	U
	34 35	—	—	—	—	—	—
	36 37	—	—	—	—	—	—
S.IN	38 39	α	Filter coefficient	0 to 1	—	U	U
S.PHPL	40 41	HS	Upper/lower limit alarm hysteresis	0 to 999999	%	U	U
S.PHPL	42 43	CTIM	Change rate alarm check time	0 to 999999	s	U	U
S.PHPL	44 45	DPL	Change rate alarm value	0 to 100	%	U	U
S.ONF2/S.ONF3	46 47	CT	Control cycle	0 to 999999	s	U	U

[*1] MODE, ALM, and INH are shared among the instructions.

Appendix 2.6 Batch counter (SBC)

Instruction used	Offset	Item	Name	Recommended range	Unit	Data storage
						SBC
	+0	—	—	—	—	—
	1	MODE[*1]	Operation mode	0 to FFFF$_H$	—	S/U
	2	—	—	—	—	—
	3	ALM[*1]	Alarm detection	0 to FFFF$_H$	—	S/U
	4	INH[*1]	Alarm detection inhibition	0 to FFFF$_H$	—	S/U
	5	—	—	—	—	—
	6	—	—	—	—	—
	7	—	—	—	—	—
	8	—	—	—	—	—
	9	—	—	—	—	—
S. PSUM	10	SUM1	Integration value (Integer part)	(0 to 2147483647)	—	S
	11					
S. PSUM	12	SUM2	Integration value (Fraction part)	(0 to 2147483647)	—	S
	13					
S. BC	14	SV1	Set value 1	0 to 2147483647	—	U
	15					
S. BC	16	SV2	Set value 2	0 to 2147483647	—	U
	17					
	18	—	—	—	—	—
	19					
	20	—	—	—	—	—
	21					
	22	—	—	—	—	—
	23					
	24	—	—	—	—	—
	25					
S. BC	26	PH	Upper limit alarm set value	0 to 2147483647	—	U
	27					
	28	—	—	—	—	—
	29					
	30	—	—	—	—	—
	31					
	32	—	—	—	—	—
	33					
	34	—	—	—	—	—
	35					
	36	—	—	—	—	—
	37					
	38	—	—	—	—	—
	39					
	40	—	—	—	—	—
	41					
S. BC	42	CTIM	Change rate alarm check time	0 to 999999	s	U
	43					
S. BC	44	DPL	Change rate alarm value	0 to 2147483647	—	U
	45					
	46	—	—	—	—	—
	47					

[*1] MODE, ALM, and INH are shared among the instructions.

Appendix 2.7 Ratio control (SR)

Instruction used	Offset	Item	Name	Recommended range	Unit	Data storage
						SR
	+0	—	—	—	—	—
	1	MODE[*1]	Operation mode	0 to FFFF_H	—	S/U
	2	—	—	—	—	—
	3	ALM[*1]	Alarm detection	0 to FFFF_H	—	S/U
	4	INH[*1]	Alarm detection inhibition	0 to FFFF_H	—	S/U
	5	—	—	—	—	—
	6	—	—	—	—	—
	7	—	—	—	—	—
	8	—	—	—	—	—
	9	—	—	—	—	—
S. PHPL	10 11	PV	Process value	(RL to RH)	—	S
S. OUT2	12 13	MV	Manipulated value	-10 to 110	%	S/U
S. R	14 15	SPR	Set value	-999999 to 999999	—	U
S. R	16 17	BIAS	Bias	-999999 to 999999	%	U
S. OUT2	18 19	MH	Output upper limit value	-10 to 110	%	U
S. OUT2	20 21	ML	Output lower limit value	-10 to 110	%	U
S. PHPL	22 23	RH	Engineering value upper limit	-999999 to 999999	—	U
S. PHPL	24 25	RL	Engineering value lower limit	-999999 to 999999	—	U
S. PHPL	26 27	PH	Upper limit alarm set value	RL to RH PL < PH	—	U
S. PHPL	28 29	PL	Lower limit alarm value	RL to RH PL < PH	—	U
S. PHPL	30 31	HH	Upper upper limit alarm value	RL to RH PH ≤ HH	—	U
S. PHPL	32 33	LL	Lower lower limit alarm value	RL to RH LL ≤ PL	—	U
	34 35	—	—	—	—	—
	36 37	—	—	—	—	—
S. IN	38 39	α	Filter coefficient	0 to 1	—	U
S. PHPL	40 41	HS	Upper/lower limit alarm hysteresis	0 to 999999	%	U
S. PHPL	42 43	CTIM	Change rate alarm check time	0 to 999999	s	U
S. PHPL	44 45	DPL	Change rate alarm value	0 to 100	%	U
S. R	46 47	CT	Control cycle	0 to 999999	s	U

Instruction used	Offset	Item	Name	Recommended range	Unit	Data storage
						SR
S. OUT2	48 49	DML	Output change rate limit value	0 to 100	%	U
S. R	50 51	DR	Change rate limit value	0 to 999999	—	U
S. R	52 53	RMAX	Ratio upper limit value	-999999 to 999999	—	U
S. R	54 55	RMIN	Ratio lower limit value	-999999 to 999999	—	U
S. R	56 57	Rn	Ratio current value	(-999999 to 999999)	—	S

*1 MODE, ALM, and INH are shared among the instructions.

Appendix 3 Operation Processing Time

Appendix 3.1 Operation processing time of each instruction

The operation processing time of each instruction is indicated in the table on this page and later.

Since the operation processing time changes depending on the setting conditions, refer to the value in the table as the guideline of the processing time.

Instruction	Condition	Processing time(μs)	
		QnPHCPU/QnPRHCPU	QnUDPVCPU
S.IN	Condition where ALM does not turn ON during loop run	69	34
S.OUT1	Condition where ALM does not turn ON during loop run in AUT mode	47	30
S.OUT2	Condition where ALM does not turn ON during loop run in AUT mode	37	28
S.MOUT	Executed during loop run in MAN mode	27	20
S.DUTY	Execution cycle = 1, Control output cycle = 10 Condition where ALM does not turn ON during loop run in AUT mode	55	30
S.BC	Condition where ALM does not turn ON during loop run in AUT mode	29	21
S.PSUM	Integration start signal = ON Integration hold signal = OFF	23	17
S.PID	Set value pattern = 3(Without cascade) Tracking bit = 0 Execution cycle = Control cycle = 1 Integral constant ≠ 0 Derivative constant ≠ 0 Condition where ALM does not turn ON during loop run in AUT mode	104	47
S.2PID	Set value pattern = 3(Without cascade) Tracking bit = 0 Execution cycle = Control cycle = 1 Integral constant ≠ 0 Derivative constant ≠ 0 Condition where ALM does not turn ON during loop run in AUT mode	136	55
S.PIDP	Set value pattern = 3(Without cascade) Tracking bit = 0 Execution cycle = Control cycle = 1 Integral constant ≠ 0 Derivative constant ≠ 0 Condition where ALM does not turn ON during loop run in AUT mode	119	57
S.SPI	Set value pattern = 3(Without cascade) Tracking bit = 0 Operating time = Sample cycle (ST = STHT) Integral constant ≠ 0 Condition where ALM does not turn ON during loop run in AUT mode	87	42
S.IPD	Set value pattern = 3(Without cascade) Tracking bit = 0 Execution cycle = Control cycle = 1 Integral constant ≠ 0 Derivative constant ≠ 0 Condition where ALM does not turn ON during loop run in AUT mode	101	47
S.BPI	Set value pattern = 3(Without cascade) Tracking bit = 0 Execution cycle = Control cycle = 1 Integral constant ≠ 0 Condition where ALM does not turn ON during loop run in AUT mode	75	39
S.R	Set value pattern = 3(Without cascade) Tracking bit = 0 Execution cycle = Control cycle = 1 Executed during loop run in AUT	58	30

Instruction	Condition	Processing time(μs)	
		QnPHCPU/QnPRHCPU	QnUDPVCPU
S.PHPL	Condition where ALM does not turn ON during loop run in AUT mode	100	40
S.LLAG	Input data = 50, With lead-lag guarantee Lead time = 1, Delay time = 1	30	20
S.I	Input data = 50, Integral time = 1 Output initial value = 0	23	17
S.D	Input data = 50, Derivative time = 1 Output initial value = 0	27	19
S.DED	Input data = 50 Operation control signal 0 → 1 Data collection interval = 1 Sampling count = 10 Output initial value = 0 Initial output switching = 0	17	12
S.HS	Input number = 5 Input data = 50, 100, 150, 200, 250	29	13
S.LS	Input number = 5 Input data = 50, 100, 150, 200, 250	32	13
S.MID	Input number = 5 Input data = 50, 100, 150, 200, 250	69	19
S.AVE	Input number = 2, Input data = 50, 100	24	15
S.LIMT	Input data = 50 Upper limit value = 100 Lower limit value = 0 Upper limit hysteresis = 0 Lower limit hysteresis = 0	30	19
S.VLMT1	Input data = 50 Positive direction limit value = 100 Negative direction limit value = 100 Positive direction hysteresis = 0 Negative direction hysteresis = 0	38	19
S.VLMT2	Input data = 50 Positive direction limit value = 100 Negative direction limit value = 100 Positive direction hysteresis = 0 Negative direction hysteresis = 0	27	18
S.ONF2	Input data = 10 Set value pattern = 3(Without cascade) Tracking bit = 0 Execution cycle = Control cycle = 1 Executed during loop run in MAN mode	52	35
S.ONF3	Input data = 10 Set value pattern = 3(Without cascade) Tracking bit = 0 Execution cycle = Control cycle = 1 Executed during loop run in MAN mode	59	36
S.DBND	Input data = 50 Dead band upper limit = 100, Dead band lower limit = 0 Input range = 1	26	17
S.PGS	Number of operation constant polygon points = 16 Operation type = 0(Hold type) Execution cycle = 1 Set value = 10 Condition where ALM does not turn ON during loop run in AUT mode	86	26
S.SEL	Set value pattern = 18H(E1, E2 Used, Without cascade) Tracking bit = 0 Condition where ALM does not turn ON during loop run in AUT mode	68	34

Instruction	Condition	Processing time(μs)	
		QnPHCPU/QnPRHCPU	QnUDPVCPU
S.BUMP	Output set value = 0, Output control value = 50 Mode selection signal = 1 Delay time = 1, Delay zone = 1	18	14
S.AMR	Output addition value = 50, Output subtraction value = 50 Output set value = 0, Output signal = 1 Output addition signal = 1, Output subtraction signal = 0 Output upper limit value = 50, Output lower limit value = 0	25	16
S.FG S.IFG	Input data = 50, Number of polygon points = 2 Polygon coordinates (30, 40), (60, 70)	33	23
S.FLT	Input data = 50, Data collection interval = 1 Sampling count = 10	40	25
S.SUM	Input data = 50 Input low cut value = 0, Initial value = 0 Input range = 1	25	18
S.TPC	Both temperature and pressure are corrected. Differential pressure = 100, Measurement temperature = 300 Measured pressure = 10000, Design temperature = 0 Bias (Temperature) = 273.15 Design pressure = 0 Bias pressure = 10332.0	39	19
S.ENG S.IENG	Input data = 50, Engineering value upper limit = 100 Engineering value lower limit = 0	25	19
S.ADD	Input number = 2, Input data = 50, 100 Number of coefficients = 2, Coefficient = 1, 1, Bias = 0	25	19
S.SUB	Input number = 2, Input data = 50, 100 Number of coefficients = 2, Coefficient = 1, 1, Bias = 0	26	20
S.MUL	Input number = 2, Input data = 50, 100 Number of coefficients = 2, Coefficient = 1, 1, Bias = 0	24	19
S.DIV	Input data = 50, 100 Coefficient = 1, 1, 1, Bias = 0, 0, 0	27	18
S.SQR	Input data = 50 Output low cut value = 0, Coefficient = 10	34	17
S.ABS	Input data = 50	17	12
S.>	Input data = 50, 100 Set value = 0, Hysteresis = 0	22	16
S.<	Input data = 50, 100 Set value = 0, Hysteresis = 0	19	15
S.=	Input data = 50, 100 Set value = 0	18	15
S.>=	Input data = 50, 100 Set value = 0, Hysteresis = 0	22	16
S.<=	Input data = 50, 100 Set value = 0, Hysteresis = 0	19	15
S.AT1	Set value pattern = 3(Without cascade) Tracking bit = 0 Execution cycle = 1 Executed during loop run in MAN mode	67	30

Appendix 3.2 Operation processing time of 2-degree-of-freedom PID control loop

This section gives an example of the operation constant of each instruction and the processing times taken when actual values are stored into the loop tag memory.

(1) Conditions

- Loop type: S2PID
- Used instructions: S.IN, S.PHPL, S.2PID, S.OUT1

(2) Operation constants

(a) S.IN instruction

Name	Item	Setting
Engineering conversion upper limit	EMAX	100.0
Engineering conversion lower limit	EMIN	0.0
Input upper limit	NMAX	100.0
Input lower limit	NMIN	0.0
Upper limit range error occurrence	HH	95.0
Upper limit range error return	H	80.0
Lower limit range error return	L	20.0
Lower limit range error occurrence	LL	5.0

(b) S.PHPL instruction: Without operation constant

(c) S.2PID instruction

Name	Item	Setting
Derivative gain	MTD	4.0
Deviation large alarm hysteresis	DVLS	3.0
Operation mode	PN	0
Tracking bit	TRK	0
Set value pattern	SVPTN	3

(d) S.OUT1 instruction

Name	Item	Setting
Output conversion upper limit	NMAX	100.0
Output conversion lower limit	NMIN	0.0

(3) Loop tag memory

Offset	Item	Name	Recommended range	Setting
+0	—	—	—	0
+1	MODE	Operation mode	0 to $FFFF_H$	10_H
+2	—	—	—	0
+3	ALM	Alarm detection	0 to $FFFF_H$	0
+4	INH	Alarm detection inhibition	0 to $FFFF_H$	0
+5	—	—	—	0
+6	—	—	—	0
+7	—	—	—	0
+8	—	—	—	0
+9	—	—	—	0
+10	PV	Process value	RL to RH	0.0
+12	MV	Manipulated value	-10 to 110	0.0
+14	SV	Set value	RL to RH	55.0
+16	DV	Deviation	-110 to 110	7
+18	MH	Output upper limit value	-10 to 110	100.0
+20	ML	Output lower limit value	-10 to 110	0.0
+22	RH	Engineering value upper limit	-999999 to 999999	100.0
+24	RL	Engineering value lower limit	-999999 to 999999	0.0
+26	PH	Upper limit alarm set value	RL to RH	80.0
+28	PL	Lower limit alarm value	RL to RH	20.0
+30	HH	Upper limit alarm value	RL to RH	90.0
+32	LL	Lower limit alarm value	RL to RH	10.0
+34	—	—	—	0
+36	—	—	—	0
+38	α	Filter coefficient	0 to 1	0.0
+40	HS	Upper/lower limit alarm hysteresis	0 to 999999	3.0
+42	CTIM	Change rate alarm check time	0 to 999999	8.0
+44	DPL	Change rate alarm value	0 to 100	30.0
+46	CT	Control cycle	0 to 999999	1.0
+48	DML	Output change rate limit value	0 to 100	100.0
+50	DVL	Deviation limit value	0 to 100	25.0
+52	P	Gain	0 to 999999	3.0
+54	I	Integral constant	0 to 999999	8.0
+56	D	Derivative constant	0 to 999999	5.0
+58	GW	Gap width	0 to 100	15.0
+60	GG	Gap gain	0 to 999999	2.0
+62	MVP	MV inside operation value	-999999 to 999999	0.25
+64	α	2-degree-of-freedom parameter α	0 to 1	0.0
+66	β	2-degree-of-freedom parameter β	0 to 1	1.0

(4) Processing time

(a) Processing times of used instructions

Instruction	For QnPHCPU	For QnUDPVCPU
S.IN	69µs	34µs
S.PHPL	100µs	40µs
S.2PID	136µs	55µs
S.OUT1	47µs	30µs

(b) Processing time of loop type

Instruction	For QnPHCPU	For QnUDPVCPU
S2PID	352µs	159µs

INDEX

INSTRUCTION INDEX

REVISIONS

*The manual number is given on the bottom left of the back cover.

Print date	*Manual number	Revision
Apr., 2002	SH (NA)-080316E-A	First edition
Jun., 2004	SH (NA)-080316E-B	Manual name change QnPHCPU Programming Manual (Process Control Instructions) → QnPHCPU/QnPRHCPU Programming Manual (Process Control Instructions) Term change DVL: Change rate limit value → Deviation limit value Partial addition About Manuals, Section 2.2.1, 2.2.3, 3.3.5, Chapter 7, Section 8.5, 9.18, 9.21, 10.1, 10.2, Appendix 2.3, Appendix 2.7 Addition Generic terms and abbreviations used in this manual
May, 2005	SH (NA)-080316E-C	Correction CONTETNTS, Section 3.2, 6.2.3, 8.2, 8.5, 9.1, 9.2, 9.3, 9.4, 9.5, 9.8, 10.1, 10.2
May, 2008	SH (NA)-080316E-D	Revision due to the addition of Process CPU Addition module Q02PHCPU, Q06PHCPU Partial correction GENERIC TERM AND ABBREVIATIONS USED IN THIS MANUAL, Section 9.18
Apr., 2009	SH (NA)-080316E-E	Partial correction SAFETY CAUTIONS, ABOUT MANUALS, Section 8.1, Appendix 1
Feb., 2013	SH (NA)-080316E-F	Descriptions related to the structured ladder/FBD and structured text language are added.
Sep., 2013	SH (NA)-080316E-G	Manual name change QnPHCPU/QnPRHCPU Programming Manual (Process Control Instructions) → MELSEC-Q Programming/Structured Programming Manual (Process Control Instructions) Correction Chapter 1, Section 2.2.1, 2.2.4, 2.2.5, 2.2.8, 3.3.1, 3.3.3, CHAPTER 7 to 14, Appendix 2, Appendix 3.1, Appendix 3.2
Oct., 2018	SH (NA)-080316E-H	Correction TERMS, Chapter 8, 9, 10, 11, 12, 13, 14, Appendix 3.1, Appendix 3.2
Dec., 2019	SH(NA)-080316E-I	Correction Section 2.3, 4.2.1, 6.2.1, 8.6, 9.8, Appendix 1

Japanese Manual Version SH-080265-H

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.

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 of damages caused by any cause found not to be the responsibility of Mitsubishi, loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products, special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products, 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.

SH(NA)-080316E-I(1912)MEE
MODEL: QNPHCPU-P-PROCE-E
MODEL CODE: 13JF67

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