QCPU(Q Mode)/QnACPU

MITSUBISHI

Programming Manual

(PID Control Instructions)



Mitsubishi Programmable Controller



• SAFETY PRECAUTIONS •

(You must read these cautions before using the product)

As for the use of the product, please carefully read this manual and the related manuals introduced later. Also, please pay attention to safety adequately and manage the product correctly.

The safety cautions shown in this manual apply to the product only.

In this manual, the safety precautions are ranked as "DANGER" and "CAUTION".



Note that the \triangle CAUTION level may lead to a serious consequence according to the circumstances. Always follow the instructions of both levels because they are important to personal safety.

Please store this manual in order to read whenever it is necessary. Also, always forward this manual to the end users.

[Design Precautions]

 Install a safety circuit external to the programmable controller that keeps the entire system safe even when there are problems with the external power supply or the programmable controller. Otherwise, it may cause an output error or an operating error, resulting in an accident. (1) Configure a circuit such as an emergency stop circuit and a protective circuit on the outside of the programmable logic controller. (2) When the programmable controller detects the following problems, it will stop calculation and turn off all outputs. An overcurrent protective device or an overvoltage protective device in a power supply
module start running. • A watchdog timer error or others is detected with self-checking function in the programmable controller CPU. All outputs may be turned on, when an error occurs in the part of I/O controlling or others
that the programmable controller CPU cannot detect. Build a fail safe circuit exterior to the programmable controller to keep the entire system safe. As for the fail safe circuit, refer to a CPU module User's Manual.
 Configure a circuit that turns on an external power supply when the main power of programmable controller is turned on. If the external power supply is turned on first, it could result in an output error or an operating error.

[Design Precautions]

• When connecting a peripheral device to the CPU module or connecting a personal computer or others to an intelligent function module, always configure an interlock circuit in the sequence program to ensure that the whole system always operate safely.

Also, make sure to read this manual carefully and check all operations for safety first before executing other control (program changes, changes of operation status (and status control)) of the operating sequence.

Especially for the control described above on the remote sequence from an external device, an immediate action may not be taken for a programmable controller's trouble due to a data communication fault.

Configure the interlock circuit in the sequence program. Simultaneously a recovery method for system, in which a data communications fault occurs, should be determined between the external device and the programmable controller CPU.

[Startup/Maintenance Precaution]

 Make sure to read this manual carefully and check all operations for safety first before connecting a peripheral device to an operating CPU module online (particularly program changes, forced outputs, and changes of operation status). Otherwise, an operating error may cause damage or problems with the modules.

REVISIONS

* The manual number is given on the bottom left of the back cover.

Print Date	* Manual Number	* The manual number is given on the bottom left of the back cover. Revision
Dec., 1999	SH (NA) 080040-A	
Jun., 2001	SH (NA) 080040-B	Partial addition About Manuals, Chapter 1, Chapter 2, Section 2.1, 3.1, 3.2, 3.3, 3.3.1, 4.2.3, 4.3.2, 4.3.5, Chapter 5, Section 5.1, 5.2, Chapter 6, Chapter 7, Section 8.1, 8.2
Apr., 2002	SH (NA) 080040-C	Correction Chapter 1, Chapter 7, Section 8.1, 8.2, 8.3, 8.4, 8.5
Jan., 2003	SH (NA) 080040-D	 Addition of use of Basic model QCPU Addition of explanation of incomplete derivative Overall reexamination
Mar., 2003	SH (NA) 080040-E	 Addition of explanation of incomplete derivative to High Performance model QCPU
Dec., 2003	SH (NA) 080040-F	Correction Chapter 1
Jun., 2004	SH (NA) 080040-G	Addition of Redundant CPU Partial addition About Manuals, Chapter 1, Chapter 2, Section 2.1, 3.1.1, 3.1.3, 3.2.1, 3.2.3, 4.3.5, 5.1, 5.2, Chapter 6, Chapter 7, Section 8.1.1 to 8.1.4, Section 9.1.1 to 9.1.5, 9.2, Appendix 1
Sep., 2006	SH (NA) 080040-H	Partial addition Section 4.2.5, Appendix 2
Apr.,2007	SH (NA) 080040-I	Addition of Universal model QCPU Addition module Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU Partial correction GENERIC TERMS AND ABBREVIATIONS USED IN THIS MANUAL, Chapter 1, Chapter 2, Section 2.1, 3.1.1, 3.1.3, 3.2.1, 3.2.3, 5.1, Chapter 6, Chapter 7, 8.1.1 to 8.1.5, 9.1.1 to 9.1.5, Appendix 1
Mar.,2008	SH (NA) 080040-J	Addition of Universal model QCPU Addition module Q13UDHCPU, Q26UDHCPU Partial correction GENERIC TERMS AND ABBREVIATIONS USED IN THIS MANUAL, Section 2.1, Appendix 1

		* The manual number is given on the bottom left of the back cover.
Print Date	* Manual Number	Revision
May,2008	SH (NA) 080040-K	Revision due to the addition of Process CPU and Universal model QCPU
		Addition module
		Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q13UDEHCPU,
		Q26UDEHCPU
		Q02PHCPU, Q06PHCPU
		Partial correction
		GENERIC TERMS AND ABBREVIATIONS USED IN THIS MANUAL,
		Section 2.1, Appendix 1
L		

* The manual number is given on the bottom left of the back cover.

Japanese Manual Version SH-080022-K

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.

© 1999 MITSUBISHI ELECTRIC CORPORATION

INTRODUCTION

Thank you for choosing the Mitsubishi MELSEC-Q/QnA Series of Programmable Logic Controllers. Please read this manual carefully so that the equipment is used to its optimum. A copy of this manual should be forwarded to the end User.

CONTENTS

SAFETY CAUTIONS	A- 1
REVISIONS	A- 3
CONTENTS	A- 5
About Manuals	
Generic Terms and Abbreviations Used in This Manual	
1. GENERAL DESCRIPTION	1 – 1 to 1 - 3
1.1 PID Processing Method	1 - 3
2. SYSTEM CONFIGURATION FOR PID CONTROL	2 - 1 to 2 - 2
2.1 Applicable PLC CPU	2 - 2
3. PID CONTROL SPECIFICATIONS	3 - 1 to 3 - 14
3.1 PID Control by Incomplete derivative	3 - 1
3.1.1 Performance specifications	
3.1.2 PID operation block diagram and operation expressions	
3.1.3 PID control instruction list	
3.2 PID Control by Complete Derivative	
3.2.1 Performance specifications	
3.2.2 PID operation block diagram and operation expressions	
3.2.3 PID control instruction list	3 - 10
4. FUNCTIONS OF PID CONTROL	4 - 1 to 4 - 14
4.1 Outline of PID Control	4 - 1
4.2 Functions of PID Control.	
4.2.1 Operation method	
4.2.2 Forward operation and reverse operation	
4.2.3 Proportionate operation (P operation)	
4.2.4 Integrating operation (I operation)	
4.2.5 Differentiating operation (D operation)	
4.2.6 PID operation	
4.3 Other Functions	
4.3.1 Bumpless changeover function	
4.3.2 MV higher/lower limit control function	
4.3.3 Monitorning PID control with the AD57(S1) (QnACPU only)	
4.3.4 Function for transfer to the SV storage device for the PV in manual mode	4 - 12
4.3.5 Changing the PID control data or input/output data setting range (QCPU only)	

5. PID CONTROL PROCEDURE	5 - 1 to 5 - 24
5.1 PID Control Data	5 - 4
5.1.1 Number of loops to be used and the number of loops to be executed in a single scan.	5 - 15
5.1.2 Sampling cycle 5.2 I/O Data	
6. PID CONTROL INSTRUCTIONS	6 - 1 to 6 - 2

7. HOW TO READ EXPLANATIONS FOR INSTRUCTIONS	7 - 1 to 7 - 2

8. INCOMPLETE DERIVATIVE PID CONTROL INSTRUCTIONS AND PROGRAM EXAMPLES 8 - 1 to 8 - 16

8.1 PID Control Instructions		1
8.1.1 PID control data settings	S.PIDINIT,SP.PIDINIT	
8.1.2 PID operation	S.PIDCONT,SP.PIDCONT	3
8.1.3 Operation stop/start of designated loop no.	S.PIDSTOP,SP.PIDSTOP,S.PIDRUN,SP.PIDRUN	
		5
8.1.4 Parameter change at designated loop	S.PIDPRMW,SP.PIDPRMW8 - 6	3
8.2 PID Control Program Examples		3
8.2.1 System configuration for program examples.		3
8.2.2 Program example for automatic mode PID c	ontrol	9
8.2.3 Program example for changing the PID cont	rol mode between automatic and manual	3

9. COMPLETE DERIVATIVE PID CONTROL INSTRUCTIONS AND PROGRAM EXAMPLES 9 - 1 to 9 - 28

9.1 PID Control Instructions		9 - 1
9.1.1 PID control data settings	PIDINIT, PIDINITP	9 - 2
9.1.2 PID control	PIDCONT,PIDCONTP	9 - 3
9.1.3 Monitoring PID control status (QnACPU only) PID57,PID57P	9 - 5
9.1.4 Operation stop/start of designated loop no.	PIDSTOP, PIDSTOPP, PIDRUN, PIDRUNP	9 - 8
9.1.5 Parameter change at designated loop	PIDPRMW,PIDPRMWP	9 - 9
9.2 PID Control Program Examples (QCPU only)		9 - 11
9.2.1 System configuration for program examples.		9 - 11
9.2.2 Program example for automatic mode PID c	ontrol	9 - 12
9.2.3 Program example for changing the PID cont	rol mode between automatic and manual	9 - 15
9.3 PID Control Program Examples (QnACPU only).		9 - 19
9.3.1 System configuration for program examples.		9 - 19
9.3.2 Program example for automatic mode PID c	ontrol	9 - 20
9.3.3 Program example for changing the PID cont	rol mode between automatic and manual	9 – 24

APPENDIX	APP - 1 to APP - 3
Appendix 1 Processing Time List	APP - 1
Appendix 2 Anti-Reset Windup Measure	

About Manuals

The following manuals are also related to this product. In necessary, order them by quoting the details in the tables below.

Related Manuals

Manual Name	Manual Number (Model Code)
QCPU User's Manual (Function Explanation, Program Fundamentals) Describes the functions, programming procedures, devices, etc. necessary to create programs. (Sold separately)	SH-080484ENG (13JR73)
QnACPU Programming Manual (Fundamentals) Describes how to create programs, the names of devices, parameters, and types of program. (Sold separately)	IB-66614 (13JF46)
QCPU (Q mode) /QnACPU Programming Manual (Common Instructions) Describes how to use sequence instructions, basic instructions, and application instructions. (Sold separately)	SH-080039 (13JF58)
QnACPU Programming Manual (Special Function) Describes the dedicated instructions for special function modules available when using the Q2ACPU(S1), Q3ACPU, and Q4ACPU. (Sold separately)	SH-4013 (13JF56)
QnACPU Programming Manual (AD57 Instructions) Describes the dedicated instructions for controlling an AD57(S1) type CRT controller module available when using the Q2ACPU(S1), Q3ACPU, or Q4ACPU. (Sold separately)	IB-66617 (13JF49)

Before reading this manual, refer to the user's manual of the used CPU module or the QnACPU Programming Manual (Fundamentals), and confirm which programs, I/O processing, and devices can be used with the used CPU module.





Generic Terms and Abbreviations Used in This Manual

Conoria torm/abbraviation	Description of generic term/obbroviation
Generic term/abbreviation	Description of generic term/abbreviation
CPU module	Generic term of Basic model QCPU, High Performance model QCPU,
	Redundant CPU, Universal model QCPU, QnACPU
QnACPU	Abbreviation of Q2ASCPU, Q2ASCPU-S1, Q2ASHCPU, Q2ASHCPU-S1,
	Q2ACPU, Q2ACPU-S1, Q3ACPU, Q4ACPU, Q4ARCPU
QnA	Abbreviation of Q2ASCPU, Q2ASCPU-S1, Q2ASHCPU, Q2ASHCPU-S1,
	Q2ACPU, Q2ACPU-S1, Q3ACPU, Q4ACPU
Q4AR	Abbreviation of Q4ARCPU
	Abbreviation of Q00CPU, Q01CPU, Q02CPU, Q02HCPU, Q06HCPU,
	Q12HCPU, Q25HCPU, Q12PRHCPU, Q25PRHCPU, Q02UCPU,
QCPU	Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, Q26UDHCPU,
	Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q13UDEHCPU,
	Q26UDEHCPU
QnCPU	Abbreviation of Q02CPU
QnHCPU	Abbreviation of Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU
QnPHCPU	Abbreviation of Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU
QnPRHCPU	Abbreviation of Q12PRHCPU, Q25PRHCPU
	Abbreviation of Q03UDCPU, Q04UDHCPU, Q06UDHCPU,
QnUD(H)CPU	Q13UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU,
	Q06UDEHCPU, Q13UDEHCPU, Q26UDEHCPU
Basic model QCPU	
Basic	Generic term of Q00JCPU, Q00CPU, Q01CPU
High Performance model QCPU	
High Performance	Generic term of Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU
Process CPU	Generic term of Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU
Redundant CPU	Generic term of Q12PRHCPU, Q25PRHCPU
Universal model QCPU	Generic term of Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU,
Linivernel	Q13UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU,
Universal	Q06UDEHCPU, Q13UDEHCPU, Q26UDEHCPU

This manual uses the following generic terms and abbreviations unless otherwise described.

MEMO

1. GENERAL DESCRIPTION

This manual describes the sequence program instructions used to implement PID control with any of the following CPU modules.

- · Basic model QCPU (first five digits of serial No. are 04122 or later)
- High Performance model QCPU
- Redundant CPU
- Universal model QCPU
- QnACPU

The Basic model QCPU, High Performance model QCPU, Redundant CPU, and Universal model QCPU have the instructions used to perform PID control by incomplete derivative (PID control instructions) and the instructions used to perform PID control by complete derivative (PID control instructions) as standard features.

The QnACPU has the instructions used to perform PID control by complete derivative (PID control instructions) as standard features.

Since the incomplete derivative PID control instructions and complete derivative PID control instructions are independent of each other, they can be executed at the same time.

The following table indicates the CPU modules that can use the incomplete derivative PID control instructions and complete derivative PID control instructions.

CPU Moo	Incomplete Derivative	Complete Derivative	
Basic model QCPU	First five digits of serial No. are "04121" or earlier	×	×
Basic model QCF U	First five digits of serial No. are "04122" or later	0	0
High Performance model QCPU	First five digits of serial No. are "05031" or earlier	×	0
	First five digits of serial No. are "05032" or later	^O *1	0
Redundant CPU		0	0
Universal model QCPU	0	0	
QnACPU	×	0	

 \bigcirc : Usable, \times : Unusable

*1: Version 7 or earlier version of GX Developer issues an "instruction code alarm" if it loads a new CPU instruction realized with GX Developer Version 8.

There are the following PID control instructions.

Classification	Incomplete Derivative	Complete Derivative
PID control data setting	S(P).PIDINIT	PIDINIT(P)
PID operation	S(P).PIDCONT	PIDCONT(P)
PID control status monitor	—	PID57(P)
Specified loop No. operation stop	S(P).PIDSTOP	PIDSTOP(P)
Specified loop No. operation start	S(P).PIDRUN	PIDRUN(P)
Specified loop No. parameter change	S(P).PIDPRMW	PIDPRMW(P)

PID control via PID control instructions is implemented by combining the CPU module with the A/D converter module and D/A converter module.

In the case of the QnACPU, the PID control status can be monitored using the AD57(S1) CRT controller module.

POINT

(1) The Process CPU is not compatible with the PID control instructions described in this manual.

To implement PID control using the Process CPU, use the process control instructions described in the QnPHCPU/QnPRHCPU Programming Manual (Process Control Instructions).

(2) The Redundant CPU can use the PID control instructions and process control instructions.

1.1 PID Processing Method

This section describes the processing method for PID control using PID control instructions. (For details on PID operations, see Chapter 4.)

Execute PID control with PID control instructions by loading an A/D converter module and a D/A converter module, as shown in Figure 1.1.



Figure 1.1 Overview of PID Control Processing

In the PID control processing method, as shown in Figure 1.1, the PID operation is executed using the set value (SV) and the process value (PV) read from the A/D converter module, and the manipulated value (MV) is then calculated.

The calculated MV (manipulated value) is output to the D/A converter module.

When a PID operation instruction* is executed in a sequence program, the sampling cycle is measured and a PID operation is performed.

PID operation in accordance with the PID operation instruction is executed in preset sampling cycles.



Figure 1. 2 Operation when PID Operation Instruction Executed

REMARK

*: There are the following PID operation instructions.

- S.PIDCONT (incomplete derivative)
- PIDCONT (complete derivative)

MEMO

-	

2. SYSTEM CONFIGURATION FOR PID CONTROL

This chapter describes the system configuration for PID control using the PID control instructions.

For the modules that can be used to configure a system, refer to the following manual. • Basic model QCPU, High Performance model QCPU, Universal model QCPU: MELSEC-Q DATA

BOOK

• QnACPU: User's manual (details) of the used CPU module



POINT

SV, PV and MV used with the PID control instructions may be set either with the fixed values of 0 to 2000 or to any values according to the used module. Refer to Section 4.3.5 for details.

CPU Module Type	SV, PV, MV			
CFO Module Type	0 to 2000 fixed *	Any setting		
Basic model QCPU	0	0		
High Performance model QCPU	0	0		
Redundant CPU	0	0		
Universal model QCPU	0	0		
QnACPU	0	×		
	O: Can b	e set, $ imes$: Cannot be set		
Vhen the resolution of the A/D con	verter module or D/A	converter module used for		
O of PID control is other than 0 to	2000, convert the digi	ital values into 0 to 2000.		

2.1 Applicable PLC CPU

Component	Module
Basic model QCPU	Q00JCPU, Q00CPU, Q01CPU
	(First 5 digits of serial No. are 04122 or later)
High Performance model QCPU	Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU
Redundant CPU	Q12PRHCPU, Q25PRHCPU
	Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU,
Universal model QCPU	Q13UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU,
	Q06UDEHCPU, Q13UDEHCPU, Q26UDEHCPU
	Q2ASCPU, Q2ASCPU-S1, Q2ASHCPU, Q2ASHCPU-S1
QnACPU	Q2ACPU, Q3ACPU, Q4ACPU, Q4ARCPU

3. PID CONTROL SPECIFICATIONS

This section gives the specifications PID operation using PID control instructions.

3.1 PID Control by incomplete derivative

3.1.1 Performance specifications

			Specifications					
				D limits	Without PID limits			
ltem			Basic model QCPU	High Performance model QCPU, Redundant CPU, Universal model QCPU	Basic model QCPU	High Performance model QCPU, Redundant CPU, Universal model QCPU	QnA CPU	
Number of	r of PID control loops - 8 loops 32 loops 8 loops 32 loops (maximum) (maximum) (maximum)				32 loops (maximum)	_		
Sampling	cycle	Ts		0.01 to	60.00 s		_	
PID opera	tion method	_	Process value differentiation incomplete derivative (forward operation/reverse operation)			_		
PID	Proportional constant	KΡ		0.01 to	100.00		_	
constant	Integral constant	Τı		0.1 to 3	000.0 s		_	
setting	Derivative constant	TD	0.00 to 300.00 s			_		
range	range Derivative gain KD		0.00 to 300.00					
SV (set va	SV (set value) setting range SV		0 to :	2000	-32768	to 32767	_	
		PV	-50 to 2050 -32768 to 32767		to 32767	-]		
MV (manip	oulated value) output range	MV						

The performance specifications for PID control are tabled below.

-: Unusable

3

3 - 1

3.1.2 PID operation block diagram and operation expressions



(2) The operation expressions for PID control using PID control instructions are indicated below.

Nam	e	Operation Expressions	Meanings of Symbols
Process	Forward operation	$\begin{split} & EV_n = PV_{fn}^* - SV \\ & \Delta MV = K_P\{(EV_n - EV_{n-1}) + \frac{T_S}{T_1} EV_n + D_n\} \\ & D_n = \frac{T_D}{T_S + \frac{T_D}{K_D}} (PV_{fn-2} PV_{fn-1} + PV_{fn-2}) + \frac{T_D}{T_S + \frac{T_D}{K_D}} D_{n-1} \\ & MV_n = \Sigma \Delta MV \end{split}$	EVn : Deviation in the present sampling cycle EVn-1 : Deviation in the preceding sampling cycle SV : Set value PVfn : Process value of the present sampling cycle (after filtering) PVfn-1 : Process value of the preceding cycle (after filtering)
value differentiation Incomplete derivative		$EV_{n}=SV-PV_{fn}^{*}$ $\Delta MV=K_{p}\{(EV_{n}-EV_{n-1})+\frac{T_{S}}{T_{1}}EV_{n}+D_{n}\}$ $D_{n}=\frac{T_{D}}{T_{S}+\frac{T_{D}}{K_{D}}}(-PV_{fn}+2PV_{fn-1}-PV_{fn-2})+\frac{T_{D}}{T_{S}+\frac{T_{D}}{K_{D}}}D_{n-1}$ $MV_{n}=\Sigma \Delta MV$	sampling cycle (after filtering) PVfn-2 : Process value of the sampling cycle two cycles before (after filtering) △MV : Output change value MVn : Present manipulation value Dn : Present derivative term Dn-1 : Derivative term of the preceding sampling cycle KP : Proportional constant Ts : Sampling cycle Ti : Integral constant TD : Derivative constant KD : Derivative gain

POINT					
(1) *:PVfn is calcu	lated using the following expression.				
Therefore, i	t is the same as the PV (process value) of the input data as long				
as the filter	coefficient is not set for the input data.				
Process	Value after Filtering $PV_{fn} = PV_n + \alpha (PV_{fn-1} - PV_n)$				
PVn	PVn : Process value of the present sampling cycle				
α	: Filter coefficient				
PV _{fn-1} : Process value of the preceding sampling cycle (after filtering)					
(2) PV _{fn} is stored	l in the I/O data area. (See Section 5.2)				

3.1.3 PID control instruction list

Instruction	Dragonaing Details	CPU		
Name	Processing Details	QCPU	QnACPU	
S.PIDINIT	Sets the reference data for PID operation.	○*	×	
S.PIDCONT	Executes PID operation with the SV (set value) and the PV (process value).	()*	×	
S.PIDSTOP S.PIDRUN	Stops or starts PID operation for the set loop No.	0	×	
S.PIDPRMW	Changes the operation parameters for the designated loop number to PID control data.	()★	×	

A list of the instructions used to execute PID control is given below.

○: Usable, ×: Unusable

*: The Basic model QCPU, High Performance model QCPU, Redundant CPU and Universal model QCPU allow selection of "with/without PID limits". Refer to Sections 5.1 and 5.2 for details of the setting range when "with/without PID limits" has been selected.

(1) PID control instruction list

The PID control instruction list has the format indicated below:



Table 3.1 How to Read the PID control Instruction List

Explanation

- (1) Classification of instructions according to their application.
- (2) Instruction names written in a sequence program.
- (3) Symbols used in the ladder diagram.
- (4) Processing for each instruction.



Fig. 3.1 Processing for Each Instruction

Symbol	Execution Condition
	Indicates an instruction that is executed for the duration that the condition for its execution is ON. When the condition before the instruction is OFF, the instruction is not executed and no processing is carried out.
	Indicates an instruction that is executed once only at the leading edge (OFF to ON) of the condition for its execution; thereafter the instruction will not be executed, and no processing will be carried out, even if the condition is ON.

(5) The execution condition for each instruction. Details are given below.

(6) Number of instruction steps

For details on the number of steps, refer to the QCPU (Q mode) /QnACPU Programming Manual (Common Instructions).

- (7) A circle indicates that subset processing is possible.
 - indicates that subset processing is impossible.

For details on subset processing, refer to the QCPU (Q mode) /QnACPU Programming Manual (Common Instructions).

(8) Indicates the page number in this manual where a detailed description for the instruction can be found.

A PID control instruction list is given in Table 3.2.

Table 3.2 PID Control Instruction List

Category	Instruction Symbol	Ladder Format	Processing Details	Execution Condition	Number of Basic Steps	Subset Processing	Page
PID Control		- S.PIDINITS-	Sets the PID control data stored in the word device (designated by (§)). (§) + 0 (§) + 1 (§) + 2 to (§) + 15 For loop 1				
data setting	S.PIDINIT	-SP.PIDINITS-	$ \begin{array}{c c} & & & & \\ \hline & & & \\ & & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & $		7		8-2
PID operation	S.PIDCONT	- <u>S.PIDCONT</u> S	Executes PID operation with the SV (set value) and the PV (process value) designated by (s) and stores the PID operation results in the MV (manipulated value) area of the word device designated by (s). (s) Common data setting area (s) + 10 (s) +		7		8-3
Operation stop	S.PIDSTOP	-s.pidstop	Stops the PID operation at the loop number designated by (n).		7	_	8-5
Operation start	S.PIDRUN	- S.PIDRUN (M)-	Starts the operation at the loop number designated by n.		6	_	8-5
Parameter change	S.PIDPRMW	- S.PIDPRMW (D)S-	Changes the operation parameter for the loop number designated by (n) to the PID control data stored in the word device designated by (S)		8	_	8-6

POINT

- (1) "PID operation by incomplete derivative" and "PID operation by complete derivative" can be executed simultaneously since they are independent.
- (2) When the S(P).PIDINIT instruction has been used to make initialization, use the S(P).PIDCONT instruction to perform PID operation.
 - To stop and start the PID operation of the specified loop No. and to change the PID control data, use the S(P).PIDSTOP, S(P).PIDRUN and S(P).PIDPRMW instructions accordingly.

3.2 PID Control by Complete Derivative

3.2.1 Performance specifications

1								
				Specification				
			With PID limits		Without PID limits			
	ltem			High Performance model QCPU,		High Performance model QCPU,		
			Basic model QCPU	Redundant CPU, Universal model QCPU	Basic model QCPU	Redundant CPU, Universal model QCPU	QnACPU	
Number of PID control loops		_	8 loops (maximum)	32 loops (maximum)	8 loops (maximum)	32 loops (maximum)	32 loops (maximum)	
Sampling	cycle	Ts	0.01 to 60.00 s					
	ion method	_	Process value differentiation complete derivative (forward operation/reverse operation)					
PID	Proportional constant	KΡ	0.01 to 100.00					
constant	Integral constant	Τı		0.1 to 3000.0 s				
setting range Derivative constant T		T⊳	0.00 to 300.00 s					
SV (set value) setting range SV			0 to	2000	-32768	to 32767	0 to 2000	
PV (process value) setting range		PV	50 to	2050	00700 1 00707		50 to 2050	
MV (manip	MV (manipulated value) output range		-50 to 2050		-32768 to 32767		-50 to 2050	

The performance specifications for PID control are tabled below.

3.2.2 PID operation block diagram and operation expressions



(2) The operation expressions for PID operation using PID control instructions are indicated below.

Name		Operation Expressions	Meanings of Symbols
Process value differentiation		$EV_{n}=PV_{fn}^{*}-SV$ $\Delta MV=K_{P}\{(EV_{n}-EV_{n-1})+\frac{T_{S}}{T_{1}}EV_{n}+D_{n}\}$ $D_{n}=\frac{T_{D}}{T_{S}} \cdot (PV_{fn}-2PV_{fn-1}+PV_{fn-2})$ $MV_{n}=\Sigma \Delta MV$	 EVn : Deviation in the present sampling cycle EVn-1 : Deviation in the preceding sampling cycle SV : Set value PVm : Process value of the present sampling cycle (after filtering) PVfn-1 : Process value of the preceding sampling cycle (after filtering) PVfn-2 : Process value of the sampling cycle two
Complete		$EV_n=SV-PV_{fn}^*$ $\Delta MV=K_P\{(EV_n-EV_{n-1})+\frac{T_S}{T_1}EV_n+D_n\}$ $D_n=\frac{T_D}{T_S}(-PV_{fn}+2PV_{fn-1}-PV_{fn-2})$ $MV_n=\Sigma \Delta MV$	$\begin{array}{l} \label{eq:state-of-the-sampling-cycle two-cycles before (after filtering) \\ \begin{tabular}{lllllllllllllllllllllllllllllllllll$

POINT						
(1) *:PV _{fn} is ca	(1) *:PV _{fn} is calculated using the following expression.					
Therefore	Therefore, it is the same as the PV (process value) of the input data as long					
as the filt	er coefficient is not set for the input data.					
Proces	Process Value after Filtering $PV_{fn} = PV_n + \alpha (PV_{fn-1} - PV_n)$					
PVn : Process value of the present sampling cycle						
α	: Filter coefficient					
PV _{fn} -	1 : Process value of the preceding sampling cycle (after filtering)					
(2) PV _{fn} is stored in the I/O data area. (See Section 5.2)						

(1) The PID operation block diagram for complete derivative is shown below.

3.2.3 PID control instruction list

Instruction	Dracessing Details	CPU			
Name	Processing Details	QCPU	QnACPU		
PIDINIT	Sets the reference data for PID operation.	` *	0		
PIDCONT	Executes PID operation with the SV (set value) and the PV (process value).	⊖*	0		
PID57	Used to monitor the results of PID operation at an AD57(S1).	×	0		
PIDSTOP PIDRUN	Stops or starts PID operation for the set loop No.	0	0		
PIDPRMW	Changes the operation parameters for the designated loop number to PID control data.	· *	0		

A list of the instructions used to exe	ecute PID control is given below.

○: Usable, ×: Unusable

*: The Basic model QCPU, High Performance model QCPU, Redundant CPU and Universal model QCPU allow selection of "with/without PID limits". Refer to Sections 5.1 and 5.2 for details of the setting range when "with/without PID limits" has been selected.

(1) The PID control instruction list

The PID control instruction list has the format indicated below:



Table 3.3 How to Read the PID control Instruction List

Explanation

- (1) Classification of instructions according to their application.
- (2) Instruction names written in a sequence program.
- (3) Symbols used in the ladder diagram.
- (4) Processing for each instruction.



Fig. 3.2 Processing for Each Instruction

Symbol	Execution Condition
	Indicates an instruction that is executed for the duration that the condition for its execution is ON. When the condition before the instruction is OFF, the instruction is not executed and no processing is carried out.
	Indicates an instruction that is executed once only at the leading edge (OFF to ON) of the condition for its execution; thereafter the instruction will not be executed, and no processing will be carried out, even if the condition is ON.

(5) The execution condition for each instruction. Details are given below.

(6) Number of instruction steps

For details on the number of steps, refer to the QCPU (Q mode) /QnACPU Programming Manual (Common Instructions).

- (7) A circle indicates that subset processing is possible.
 - indicates that subset processing is impossible.

For details on subset processing, refer to the QCPU (Q mode) /QnACPU Programming Manual (Common Instructions).

(8) Indicates the page number in this manual where a detailed description for the instruction can be found.

A PID control instruction list is given in Table 3.4.

Table 3.4 PID Control Instruction List

Category	Instruction Symbol	Ladder Format	Processing Details	Execution Condition	Number of Basic Steps	Subset Processing	Page
PID control data setting	PIDINIT	- PIDINIT S	Sets the PID control data stored in the word device (designated by (§)). (§+ 0 (§+ 1 (§+ 2 to (§+ 11) For loop 1				
		- PIDINITP S-	$ \begin{array}{c c} (\$ + 11 \\ (\$ + 12 \\ to \\ (\$ + 21 \end{array} \\ \hline For \ loop \ 2 \\ \hline \\ (\$ + (m+0) \\ to \\ (\$ + (m+9) \\ m=(n-1) \times 10+2 \end{array} $		2	_	9-2
PID operation	PIDCONT	-PIDCONT S	Executes PID operation with the SV (set value) and the PV (process value) designated by (\$) and stores the PID operation results in the MV (manipulated value) area of the word device designated by (\$). (\$) + 0 (\$) + 9 (\$) + 10 (\$) + 9 (\$) + 9 (\$) + 10 (\$) + 10 (2		9-3
Monitoring	PID57	– PID57 ଲିକ୍ତିଡି–	Monitors the PID operation results for the AD57 (S1) (designated by ①). ① : First I/O number of the AD57(S1) ⑤ : Monitor screen number			9-5	
		- PID57P n 90	1:Loop 1 to loop 8 2:Loop 9 to loop16 3:Loop17 to loop24 4:Loop25 to loop32 2 : Initial screen display request		+		

Category	Instruction Symbol	Ladder Format	Processing Details	Execution Condition	Number of Basic Steps	Subset Processing	Page
Operation stop	PIDSTOP		Stops the PID operation at the loop number designated by (n).		2		9-8
Operation start	PIDRUN		Starts the operation at the loop number designated by n.		2	_	9-8
Parameter change	PIDPRMW	- PIDPRMW	Changes the operation parameter for the loop number designated by (n) to the PID control data stored in the word device designated by (S)		3	_	9-9
		- PIDPRMWP ns					

Table 3.4 PID Control Instruction List

POINT

- (1) "PID operation by incomplete derivative" and "PID operation by complete derivative" can be executed simultaneously since they are independent.
- (2) When the PIDINIT(P) instruction was used to make initialization, use the PIDCONT(P) instruction to perform PID operation.To stop and start the PID operation of the specified loop No. and to change the

PID control data, use the PIDSTOP(P) instruction, PIDRUN(P) instruction and PIDPRMW(P) instruction.

4. FUNCTIONS OF PID CONTROL

This chapter describes PID control performed using the PID control instructions.

4.1 Outline of PID Control

PID control is applicable to process control in which factors such as flowrate, velocity, air flow volume, temperature, tension, mixing ratio, etc. must be controlled. The control for maintaining the control object at the preset value is shown in the diagram below:



Fig. 4.1 Application of PID Control Process Control

During PID control, the value measured by the sensor (process value) is compared with the preset value (set value). The output value (manipulated value) is then adjusted in order to eliminate the difference between the process value and the set value. The MV (manipulated value) is calculated by combining the proportional operation (P), the integral operation (I), and the derivative operation (D) so that the PV is brought to the same value as the SV quickly and precisely.

The MV is made large when the difference between the PV and the SV is large so as to bring the PV close to the SV quickly. As the difference between the PV and the SV gets smaller, a smaller MV is used to bring the PV to the same value as the SV gradually and accurately.

4.2 Functions of PID Control

The operation methods for PID control with the PID control instructions are the velocity type and process value derivative type. The following describes the control executed for both of these methods:

4.2.1 Operation method

(1) Velocity type operation

The velocity type operation calculates amounts of changes in the MVs (manipulated values) during PID operation. The actual MV is the accumulated amount of change of the MV calculated for each sampling cycle.

(2) Process value derivative type operation
 The process value derivative type operation executes PID operations by differentiating the PV (process value).
 Because the deviation is not subject to differentiation, sudden changes in the output due to differentiation of the changes in the deviation generated by changing the set value can be reduced.

4.2.2 Forward operation and reverse operation

Either forward operation or reverse operation can be selected to designate the direction of PID control.

- (1) In forward operation, the MV (manipulated value) increases as the PV (process value) increases beyond the SV (set value).
- (2) In reverse operation, the MV increases as the PV decreases below the SV.
- (3) In forward operation and reverse operation, the MV becomes larger as the difference between the SV and the PV increases.
- (4) The figure below shows the relationships among forward operation and reverse operation and the MV, the PV, and the SV.


(5) The figure below shows examples of process control with forward operation and reverse operation:



4.2.3 Proportional operation (P operation)

The control method for proportional operation is described below.

- (1) In proportional operation, an MV (manipulated value) proportional to the deviation (the difference between the set value and process value) is obtained.
- (2) The relationship between E (deviation) and the MV is expressed by the following formula:

MV=Kp • E

Kp is a proportional constant and is called the "proportional gain".

Condition	Proportional Operation	
When proportional gain Kp is smaller	Control operation gets slower.	
When proportional gain Kp is	Control operation gets faster.	
larger	However, hunting is more likely to occur.	

(3) The proportional operation in step response with a constant E (deviation) is illustrated in Fig. 4.2.



Fig. 4.2 Proportional Operation with a Constant Deviation

(4) A certain error produced relative to a set value is called an offset. An offset is produced in proportional operation.



4.2.4 Integral operation (I operation)

The control method for integral operation is described below.

(1) In the integral operation, the MV (manipulated value) changes continuously to zero deviation when it occurs.

This operation can eliminate the offset that is unavoidable in proportional operation.

(2) The time required for the MV in integral operation to reach the MV for proportional operation after the generation of deviation is called the integral time. Integral time is expressed as T₁.

Condition	Integral Operation	
M/hon integral time T is	Integrating effect increases and the time to	
When integral time T ₁ is	eliminate the offset becomes shorter.	
shorter	However, hunting is more likely to occur.	
When integral time T ₁ is longer	Integrating effect decreases and the time to	
	eliminate the offset becomes longer.	

(3) The integral operation in step response with a constant E (deviation) is illustrated in Fig. 4.3.





(4) Integral operation is always used in combination with proportional operation (PI operation) or with proportional and derivative operations (PID operation).
 Integral operation cannot be used independently.

4.2.5 Derivative operation (D operation)

The control method for derivative operation is described below.

- In derivative operation, an MV (manipulated value) proportional to the deviation change rate is added to the system value to zero deviation when it occurs. This operation prevents significant fluctuation at the control objective due to external disturbances.
- (2) The time required for the MV in the derivative operation to reach the MV for the proportional operation after the generation of deviation is called the derivative time. Derivative time is expressed as T_D.

Condition	Derivative Operation	
When derivative time T _D is shorter	Differentiating effect decreases.	
When derivative time T₀ is longer	Differentiating effect increases. However, hunting of short cycle is more likely to occur.	

(3) The derivative operation when the deviation is a constant value stepped response is shown in Fig. 4.4.





 (4) Derivative operation is always used in combination with proportional operation (PD operation) or with proportional and integral operations (PID operation). Derivative operation cannot be used independently.

REMARK

About the differences between complete derivative and incomplete derivative [Incomplete derivative]

Incomplete derivative is PID control that has a primary delay filter in the input of a derivative term.

The S.PIDCONT instruction is the incomplete derivative PID control instruction. Incomplete derivative is effective for the following cases.

- · Control susceptible to high-frequency noise
- When energy effective to actuate an operation end is not given when a step change occurs in a complete derivative system

[Complete derivative]

Complete derivative is PID control that uses the input of a derivative term as it is. The PIDCONT instruction is the complete derivative PID control instruction.



4.2.6 PID operation

The control method when proportional operation (P operation), integral operation (I operation), and derivative operation (D operation) are used in combination is described below.

- (1) During PID operation, the system is controlled by the MV (manipulated value) calculated in the (P + I + D) operation.
- (2) PID operation in step response with a constant E (deviation) is illustrated in Fig. 4.5.



Fig. 4.5 PID Operation with Constant Deviation

4.3 Other Functions

During PID control using the PID control instructions, MV upper/lower limit control is automatically executed by the bumpless changeover function explained below.

4.3.1 Bumpless changeover function

- (1) This function controls the MV (manipulated value) continuously when the control mode is changed between manual and automatic.
- (2) When the mode is changed (between manual and automatic), data is transferred between the "MV area for automatic mode (automatic MV)" and "MV area for manual mode (manual MV)" as described below.
 - The control mode is changed in the I/O data area (see Section 5.2).
 - (a) Changing from the manual The MV in the manual mode is transmitted to mode to the automatic mode the MV area for the automatic mode.
 - (b) Changing from the automatic The MV in the automatic mode is transmitted mode to the manual mode to the MV area for the manual mode.

POINT

- (1) Manual and automatic modes of PID control:
 - 1) Automatic mode
 - PID operation is executed with a PID control instruction.
 - The control object is controlled according to the calculated MV.
 - Manual mode
 PID operation is not executed. The MV is calculated by the user and the control object is controlled according to the user-calculated MV.
- (2) The loop set in the manual mode stores the PV (process value) in the set value area every sampling cycle.

4.3.2 MV upper/lower limit control function

- (1) The MV upper/lower limit control function controls the upper or lower limit of the MV calculated in the PID operation. This function is only effective in the automatic mode. It cannot be executed in the manual mode.
- (2) By setting the MV upper limit (MVHL) and the MV lower limit (MVLL), the MV calculated in the PID operation can be controlled within the range between the limits.



Fig. 4.6 Operation in Accordance with the MV Upper/Lower limit

(3) When the MV upper/lower limit control is used, the MV is controlled as illustrated above.

A MVHL (MV upper limit) and MVLL (MV lower limit) takes on a value between -50 and 2050 or a user-defined value (except the QnACPU).

The following are the default settings:

- Upper limit2000 (Or user-defined value)
- Lower limit0 (Or user-defined value)

The value set for the upper limit must not be smaller than the value set for the lower limit.

An error will occur if it is.

4.3.3 Monitoring PID control with the AD57(S1) (QnACPU only)

The PID control operation results can be monitored in a bar graph with an AD57(S1) CRT controller unit.

(1) The monitor screen displays the monitored information of eight loops beginning with the designated loop number.



POINT
The SV, PV, and MV present value are displayed as percentages of 2000.
1) SV percentage display $\frac{SV}{2000} \times 100$ (%)
2) PV percentage display $\frac{100}{2000} \times 100$ (%)
3) MV percentage display $\frac{MV}{2000} \times 100$ (%)

(2) Use the PID57 instruction to execute monitoring with an AD57(S1). See Section 9.1.3 for details on the PID57 instruction.

4.3.4 Function for transfer to the SV storage device for the PV in manual mode

When using the PID control instruction to perform PID control, execute the PID operation instruction also in the manual mode.

In the manual mode, it is possible to select whether the PV imported from the A/D converter module is transferred to the SV storage device or not when the PID operation instruction is executed, depending on the ON/OFF status of the PID bumpless processing flag (SM774, SM794).

PID Bumpless Processing Flag				
SM794	SM774	Operation		
(Incomplete derivative)	(Complete derivative)			
OFF		 The PV is transferred to the SV storage device when the PID operation instruction is executed. When the manual mode is switched to the automatic mode, the MV output in the manual mode is continued. When the SV is changed after switching to the automatic mode, control is performed to achieve the SV, starting from the MV output in the manual mode. 		
ON		 The PV is not transferred to the SV storage device when the PID operation instruction is executed. When the manual mode is switched to the automatic mode, control is performed to achieve the SV, starting from the MV output in the manual mode. Before switching to the automatic mode, store the SV into the SV storage device. 		

POINT

Depending on whether SM774/SM794 is ON or OFF, there are the following differences in control when the manual mode is switched to the automatic mode.

• When SM774/SM794 is OFF, the PV is transferred to the SV storage device. Therefore, there is no difference between the PV and SV when the manual mode is switched to the automatic mode.

Hence, an abrupt change does not occur in MV at the time of mode switching. Instead, since the SV after mode switching differs from the target value in the automatic mode, the user should change the SV to the target value step by step in the sequence program.

• When SM774/SM794 is ON, the PV is not transferred to the SV storage device. Therefore, there is a difference between the PV and SV when the manual mode is switched to the automatic mode.

If the difference is large at the time of mode switching, an abrupt change may occur in MV.

Use this method in a system where the mode is switched when the PV has fully neared the SV.

PID control in the automatic mode can be executed immediately without the SV being changed step by step in the sequence program.

REMARK

 The SV and PV are stored into the devices specified in the I/O data area with the PID operation instruction.

4.3.5 Changing the PID control data or input/output data setting range (QCPU only)

The setting ranges of the following data of the PID control data (refer to Section 5.1) and I/O data (refer to Section 5.2) can be changed as desired by user setting.

Item	Set Data	
	MV lower limit value	
PID control data	MV upper limit value	
PID control data	MV change rate limit value	
	PV change rate limit value	
	SV	
	PV	
I/O data	Automatic MV	
	PV after filtering	
	Manual MV	

To make the user setting valid, turn the bit corresponding to the relevant loop of the PID limit setting special register (SD774, SD775, SD794, SD795) to "1".

PID Limit Setting	Special Register	Sotting Pango			
Incomplete derivative	Complete derivative	Setting Range			
		b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0			
		0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1			
SD794	SD774				
		LOOP 16 LOOP 2			
		b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0			
		0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1			
SD795	SD775				
		LOOP 32 LOOP 18			
0: With PID limit (system fixed value)					

0: With PID limit (system fixed value) 1: Without PID limit (user setting)

POINT		
The Basic r	odel QCPU has 8 loops.	
b0 to b7 of	D774 and SD794 are valid.	

MEMO

5

5. PID CONTROL PROCEDURE

The programming procedure required to execute PID control is shown below.



- *: The following instructions are available as the PID control data setting instructions.
 - S.PIDINIT (incomplete derivative)
 - PIDINIT (complete derivative)



POINT

- Registering or changing the PID control data per sequence program scan will present no problem.
 - However, execute the the PID control data setting instructions *² when you registered or changed the PID control data.
 - If you do not execute the PID control data setting instructions instruction, the data registered or the correction made to the PID control data will not be reflected at the execution of the the PID operation instructions.
- You need not execute the PID control data setting instructions when using the parameter change instruction *³ to change the PID control data per loop.

REMARK

- *1: The following instructions are available as the PID operation instructions.
 - S.PIDCONT (incomplete derivative)
 - PIDCONT (complete derivative)
- *2: The following instructions are available as the PID control data setting instructions.
 - S.PIDINIT (incomplete derivative)
 - PIDINIT (complete derivative)
- *3: The following instructions are available as the parameter change instructions.
 - S. PIDPRMW (incomplete derivative)
 - PIDPRMW (complete derivative)

MEMO

5.1 PID Control Data

(1) PID control data is used to set the reference values for PID operation. Store the PID control data into the CPU module with the PID control data setting^{*2} instruction before executing PID operation instruction^{*1}.

The PID control data is classified into two types, "common data for all loops" and "data for individual loops".

(a) For Basic model QCPU

\backslash				Incomplete derivative				
	Data No.	Data Item	Description	With PI Setting Range	D limits User Specification Range	Without F Setting Range	PID limits User Specification Range	
Common	1	Number of loops	Sets the number of loops for which PID operation will be executed.	1 to 8	1 to 8	1 to 8	1 to 8	
setting data	2	Number of loops in one scan	Sets the number of loops for which single PID operation will be executed when the multiple loops reaches the sampling cycle time.	1 to 8	1 to 8	1 to 8	1 to 8	
	1	Selection of operational expression	Selects the PID operational expression indicated in Section 3.1.2/Section 3.2.2.	Forward operation: 0 Reverse operation: 1	0 or 1	Forward operation: 0 Reverse operation: 1	0 or 1	
	2	Sampling cycle (Ts)	Sets the cycle of PID operation.	0.01 to 60.00 s	1 to 6000 (unit: 10 ms)	0.01 to 60.00 s	1 to 6000 (unit: 10 ms)	
	3	Proportional constant (K _P)	PID operation Proportional gain	0.01 to 100.00	1 to 10000 (unit: 0.01)	0.01 to 100.00	1 to 10000 (unit: 0.01)	
Data for each loop	4	Integral constant (Ti)	This constant expresses the magnitude of the integral operation (I operation) effect. Increasing the integral constant slows down the manipulated value change.	$\begin{array}{c} 0.1 \text{ to } 3000.0 \text{ s} \\ \\ \text{Infinite}(\infty) \\ \text{(If the setting for T_1 exceeds} \\ 3000.0 \text{ s} \end{array}$	1 to 32767 (unit: 100 ms)	$0.1 \text{ to } 3000.0 \text{ s}$ Infinite(∞) (If the setting for T ₁ exceeds 3000.0 s)	1 to 32767 (unit: 100 ms)	
	5	Derivative constant (T⊳)	This constant expresses the magnitude of the derivative operation (D operation) effect. Increasing the derivative constant causes a significant change in the manipulated value even with slight change of the control objective.	0.00 to 300.00 s	0 to 30000 (unit: 10 ms)	0.00 to 300.00 s	0 to 30000 (unit: 10 ms)	
	6	Filter coefficient (α)	Sets the degree of filtering applied to the process value. The filtering effect decreases as the value gets closer to 0.	0 to 100 %	0 to 100	0 to 100 %	0 to 100	

Table 5.1 PID Control Data List

REMARK

- *1: The following are available as the PID operation instructions.
 - S.PIDCONT (incomplete derivative)
 - PIDCONT (complete derivative)
- $^{\ast}2$: The following are available as the PID control data setting instructions.
 - S.PIDINIT (incomplete derivative)
 - PIDINIT (complete derivative)

 With	PID limits	Without	PID limits	 Processing when Set Data is Outside the Allowable Setting Range 	
Setting Range	User Specification Range	Setting Range	User Specification Range		
1 to 8	1 to 8	1 to 8	1 to 8		
1 to 8	1 to 8	1 to 8	1 to 8	An error occurs and PID operation is not executed for all loops.	
Forward operation : 0 Reverse operation: 1	0 or 1	Forward operation : 0 Reverse operation: 1	0 or 1	An error occurs and PID operation	
0.01 to 60.00s	1 to 6000 (unit: 10 ms)	0.01 to 60.00 s	1 to 6000 (unit: 10 ms)	for the corresponding loop is not executed.	
 0.01 to 100.00	1 to 10000 (unit: 0.01)	0.01 to 100.00	1 to 10000 (unit: 0.01)		
0.1 to 3000.0 s Infinite(∞) (If the setting for T _I exceeds 3000.0 s	1 to 32767 (unit: 100 ms)	0.1 to 3000.0 s Infinite(∞) (If the setting for T ₁ exceeds 3000.0 s	1 to 32767 (unit: 100 ms)	An error occurs and PID operation for the corresponding loop is not executed.	
0.00 to 300.00 s	0 to 30000 (unit: 10 ms)	0.00 to 300.00 s	0 to 30000 (unit: 10 ms)	An error occurs and PID operation for the corresponding loop is not executed.	
0 to 100%	0 to 100	0 to 100%	0 to 100		

5. PID CONTROL PROCEDURE

Table 5.1 F	PID (Control	Data	List
-------------	-------	---------	------	------

\square					Inco	mplete derivative		
$ \rangle$	Data			With PIC	limits	Without Pl	D limits	
	No.	Data Item	Description	Setting Range	User Specification Range	Setting Range	User Specification Range	
	7	MV Lower limit (MVLL)	In the automatic mode, sets the lower limit for the MV (manipulated value) calculated in PID operation. When the MV is less than the MV lower limit, the MVLL is used as the MV.	-50 to 2050	-50 to 2050	-32768 to 32767	-32768 to 32767	
	8	MV Upper limit (MVHL)	In the automatic mode, sets the upper limit for the MV calculated in PID operation. When the MV is greater than the MV upper limit, the MVHL is used as the MV.	-50 to 2050	-50 to 2050	-32768 to 32767	-32768 to 32767	
Data for each loop	9	MV change rate limit (∆ MVL)	Sets the limit for variation between the previous MV and present MV. When the MV variation is greater than the limit value, 1 is set for bit 1 (b1) of the alarm device. Does not limit the MV variation. (If the MV variation is greater than the limit value, it is used unchanged as the MV variation to calculate the MV.)	0 to 2000	0 to 2000	-32768 to 32767	-32768 to 32767	
	10	PV change rate limit (∆ PVL)	Sets the limit for variation between the previous PV and present PV. When the PV variation is greater than the limit value, 1 is set for bit 0 (b0) of the alarm device. Does not limit the PV variation. (If the PV variation is greater than the limit value, it is used unchanged as the PV variation to perform the PID operation.)	0 to 2000	0 to 2000	-32768 to 32767	-32768 to 32767	
	11	Derivative gain (K⊳)	Sets a time period (operation delay) for derivative operation. As the value is greater, the time period decreases and operation becomes closer to complete derivative.	$\begin{array}{c} 0.00 \text{ to } 300.00\\ (\text{Ideal value is}\\ 8.00)\\ \hline\\ \text{Infinite}(\infty)\\ (\text{If the setting}\\ \text{for } K_{\text{D}} \text{ exceeds}\\ 300.00\\ \end{array}$	0 to 32767 (unit: 0.01)	0.00 to 300.00 (Ideal value is 8.00) Infinite(∞) (If the setting for K _D exceeds 300.00	0 to 32767 (unit: 0.01)	

5. PID CONTROL PROCEDURE

With PID limits		Without F	PID limits	Processing when Set Data is
Setting Range	User Specification Range	Setting Range	User Specification Range	Outside the Allowable Setting Range
-50 to 2050	-50 to 2050	-32768 to 32767	-32768 to 32767	In the case of "with PID limits", PID operation is performed after conversion into the following value. • When the MVLL or MVHL value is
-50 to 2050	-50 to 2050	-32768 to 32767	-32768 to 32767	 When the MVLL of MVHL value is less than -50, "-50" is used. When the MVLL or MVHL value is greater than 2050, "2050" is used.
0 to 2000	0 to 2000	-32768 to 32767	-32768 to 32767	 In the case of "with PID limits", PID operation is performed after conversion into the following value. When the △MVL value is less than -50, it is converted into -50. When the △MVL value is greater than 2050, it is converted into 2050.
0 to 2000	0 to 2000	-32768 to 32767	-32768 to 32767	 In the case of "with PID limits", PID operation is performed after conversion into the following value. When the △PVL value is less than -50, it is converted into -50. When the △PVL value is greater than 2050, it is converted into 2050.
_			_	An error occurs and PID operation for the corresponding loop is not executed.

(b) For High Performance model QCPU, Redundant CPU, Universal model QCPU

\setminus			Incomplete derivative					
	Data			With PI	D limits	Without F	PID limits	
	No.	Data Item	Description	Setting Range	User Specification Range	Setting Range	User Specification Range	
Common	1	Number of loops	Sets the number of loops for which PID operation will be executed.	1 to 32	1 to 32	1 to 32	1 to 32	
setting data	2	Number of loops in one scan	Sets the number of loops for which single PID operation will be executed when the multiple loops reaches the sampling cycle time.	1 to 32	1 to 32	1 to 32	1 to 32	
	1	Selection of operational expression	Selects the PID operational expression indicated in Section 3.1.2/Section 3.2.2.	Forward operation: 0 Reverse operation: 1	0 or 1	Forward operation: 0 Reverse operation: 1	0 or 1	
	2	Sampling cycle (Ts)	Sets the cycle of PID operation.	0.01 to 60.00 s	1 to 6000 (unit: 10 ms)	0.01 to 60.00 s	1 to 6000 (unit: 10 ms)	
	3	Proportional constant (K _P)	PID operation proportional gain	0.01 to 100.00	1 to 10000 (unit: 0.01)	0.01 to 100.00	1 to 10000 (unit: 0.01)	
Data for each loop	4	Integral constant (Tı)	This constant expresses the magnitude of the integral operation (I operation) effect. Increasing the integral constant slows down the manipulated value change.	0.1 to 3000.0 s Infinite(∞) (If the setting for T ₁ exceeds 3000.0 s	1 to 32767 (unit: 100 ms)	0.1 to 3000.0 s Infinite(∞) (If the setting for T ₁ exceeds 3000.0 s	1 to 32767 (unit: 100 ms)	
	5	Derivative constant (T⊳)	This constant expresses the magnitude of the derivative operation (D operation) effect. Increasing the derivative constant causes a significant change in the manipulated value even with slight change of the control objective.	0.00 to 300.00 s	0 to 30000 (unit: 10 ms)	0.00 to 300.00 s	0 to 30000 (unit: 10 ms)	
	6	Filter coefficient (α)	Sets the degree of filtering applied to the process value. The filtering effect decreases as the value gets closer to 0.	0 to 100 %	0 to 100	0 to 100 %	0 to 100	

Table 5.2 PID Control Data List

	Complete			
 With PI	D limits	Without PI	D limits	Processing if Set Data is Outside the
Setting Range	User Designation Range	Setting Range	User Designation Range	Allowable Setting Range
1 to 32	1 to 32	1 to 32	1 to 32	
1 to 32	1 to 32	1 to 32	1 to 32	An error occurs and PID operation is not executed for all loops.
Forward operation : 0 Reverse operation: 1	0 or 1	Forward operation : 0 Reverse operation: 1	0 or 1	An error occurs and PID operation for the
0.01 to 60.00 s	1 to 6000 (units: 10 ms)	0.01 to 60.00 s	1 to 6000 (units: 10 ms)	corresponding loop is not executed.
0.01 to 100.00	1 to 10000 (units: 0.01)	0.01 to 100.00	1 to 10000 (units: 0.01)	
0.1 to 3000.0 s		0.1 to 3000.0 s		
Infinite(∞) (If the setting for T ₁ exceeds 3000.0 s	1 to 32767 (units: 100 ms)	Infinite(∞) (If the setting for T ₁ exceeds 3000.0 s	1 to 32767 (units: 100 ms)	An error occurs and PID operation for the corresponding loop is not executed.
0.00 to 300.00 s	0 to 30000 (units: 10 ms)	0.00 to 300.00 s	0 to 30000 (units: 10 ms)	An error occurs and PID operation for the corresponding loop is not executed.
0 to 100 %	0 to 100	0 to 100 %	0 to 100	

5. PID CONTROL PROCEDURE

Table 5.2 PID C	ontrol Data	a List
-----------------	-------------	--------

\setminus					Inco	mplete derivative	
\backslash	Data			With PID	limits	Without PI	D limits
	No.	Data Item	Description	Setting Range	User Specification Range	Setting Range	User Specification Range
	7	MV Lower limit (MVLL)	In the automatic mode, sets the lower limit for the MV (manipulated value) calculated in PID operation. When the MV is less than the MV lower limit, the MVLL is used as the MV.	-50 to 2050	-50 to 2050	-32768 to 32767	-32768 to 32767
	8	MV Upper limit (MVHL)	In the automatic mode, sets the upper limit for the MV calculated in PID operation. When the MV is greater than the MV upper limit, the MVHL is used as the MV.	-50 to 2050	-50 to 2050	-32768 to 32767	-32768 to 32767
Data for each loop	9	MV change rate limit (∆ MVL)	Sets the limit for variation between the previous MV and present MV. When the MV variation is greater than the limit value, 1 is set for bit 1 (b1) of the alarm device. Does not limit the MV variation. (If the MV variation is greater than the limit value, it is used unchanged as the MV variation to calculate the MV.)	0 to 2000	0 to 2000	-32768 to 32767	-32768 to 32767
	10	PV change rate limit (∆ PVL)	Sets the limit for variation between the previous PV and present PV. When the PV variation is greater than the limit value, 1 is set for bit 0 (b0) of the alarm device. Does not limit the PV variation. (If the PV variation is greater than the limit value, it is used unchanged as the PV variation to perform the PID operation.)	0 to 2000	0 to 2000	-32768 to 32767	-32768 to 32767
	11	Derivative gain (K⊳)	Sets a time period (operation delay) for derivative operation. As the value is greater, the time period decreases and operation becomes closer to complete derivative.	$\begin{array}{c} 0.00 \text{ to } 300.00\\ (\text{Ideal value is}\\ 8.00)\\\\ \hline \text{Infinite}(\infty)\\ (\text{If the setting}\\ \text{for } K_{\text{D}} \text{ exceeds}\\ 300.00 \end{array} \right)$	0 to 32767 (unit: 0.01)	$\begin{array}{c} 0.00 \text{ to } 300.00\\ (\text{Ideal value is}\\ 8.00)\\\\\hline \text{Infinite}(\infty)\\ (\text{If the setting}\\ \text{for } K_{\text{D}} \text{ exceeds}\\ 300.00\\ \end{array}$	0 to 32767 (unit: 0.01)

 With PID limits		Without I	PID limits	Processing when Set Data is
Setting Range	User Specification Range	Setting Range	User Specification Range	Outside the Allowable Setting Range
-50 to 2050	-50 to 2050	-32768 to 32767	-32768 to 32767	For the High Performance model QCPU "with PID limits" or the QnACPU, PID operation is performed after conversion into the following value.
-50 to 2050	-50 to 2050	-32768 to 32767	-32768 to 32767	 When the MVLL or MVHL value is less than -50, it is converted into - 50. When the MVLL or MVHL value is greater than 2050, it is converted into 2050.
0 to 2000	0 to 2000	-32768 to 32767	-32768 to 32767	 In the case of "with PID limits", PID operation is performed after conversion into the following value. When the △MVL value is less than 0, it is converted into 0. When the △MVL value is greater than 2000, it is converted into 2000.
0 to 2000	0 to 2000	-32768 to 32767	-32768 to 32767	 In the case of "with PID limits", PID operation is performed after conversion into the following value. When the △PVL value is less than 0, it is converted into 0. When the △PVL value is greater than 2000, it is converted into 2000.
—	—	—	_	An error occurs and PID operation for the corresponding loop is not executed.

(c) For QnACPU

Table 5.3 PID Control Data List

\sum	Data No.	Data Item	Description	Setting Range	User Specifi- cation Range	Processing when Set Data is Outside the Allowable Setting Range	
Commo	1	Number of loops	Sets the number of loops for which PID operation will be executed.	1 to 32	1 to 32	An error occurs and PID	
n setting data	2	Number of loops in one scan	Number of oops in one Unitial e loops reaches the sampling cycle		1 to 32	operation is not executed for all loops.	
	1	Selection of operational expression	Selects the PID operational expression indicated in Section 3.2.2.	Forward operation: 0 Reverse operation: 1	0 or 1	An error occurs and PID operation for the corresponding	
	2	Sampling cycle (Ts)	Sets the cycle of PID operation.	0.01 to 60.00 s	1 to 6000 (unit: 10 ms)	loop is not executed.	
	3	Proportional constant (K _P)	PID operation ratio	0.01 to 100.00	1 to 10000 (unit: 0.01)		
			This constant expresses the magnitude of	0.1 to 3000.0 s Infinite(∞)		An error occurs and PID	
		Integral constant (Tı)	Integral the integral operation (I operation) effect.		1 to 32767 (unit: 100 ms)	operation for the corresponding loop is not executed.	
	5 Derivative constant (Tr		This constant expresses the magnitude of the derivative operation (D operation) effect. Increasing the derivative constant causes a significant change in the manipulated value even with slight change of the control objective.	0.00 to 300.00 s	0 to 30000 (unit: 10 ms)	An error occurs and PID operation for the corresponding	
	6	Filter coefficient (\alpha)	Sets the degree of filtering applied to the process value. The filtering effect decreases as the value gets closer to 0.	0 to 100 %	0 to 100	loop is not executed.	
Data for each loop	7	MV Lower limit (MVLL)	In the automatic mode, sets the lower limit for the MV (manipulated value) calculated in PID operation. When the MV is less than the MV lower limit, the MVLL is used as the MV.	-50 to 2050	-50 to 2050	PID operation is performed after conversion into the following valu • When the MVLL or MVHL value is less than -50, it is converted into -50.	
	8	MV Upper limit (MVHL)	In the automatic mode, sets the upper limit for the MV calculated in PID operation. When the MV is greater than the MV upper limit, the MVHL is used as the MV.	-50 to 2050	-50 to 2050	 When the MVLL or MVHL value is greater than 2050, it is converted into 2050. 	
9	9	MV change rate limit (∆ MVL)	Sets the limit for variation between the previous MV and present MV. When the MV variation is greater than the limit value, 1 is set for bit 1 (b1) of the alarm device. Does not limit the MV variation. (If the MV variation is greater than the limit value, it is used unchanged as the MV variation to calculate the MV.)	0 to 2000	0 to 2000	 PID operation is performed after conversion into the following value. When the △MVL value is less than 0, it is converted into 0. When the △MVL value is greater than 2000, it is converted into 2000. 	
	10	PV change rate limit (∆ PVL)	Sets the limit for variation between the previous PV and present PV. When the PV variation is greater than the limit value, 1 is set for bit 0 (b0) of the alarm device. Does not limit the PV variation. (If the PV variation is greater than the limit value, it is used unchanged as the PV variation to perform the PID operation.)	0 to 2000	0 to 2000	 PID operation is performed after conversion into the following value. When the △PVL value is less than 0, it is converted into 0. When the △PVL value is greater than 2000, it is converted into 2000. 	

- (2) PID control data can be set in any word device number. However, all the data used for the corresponding loops must be set in devices with consecutive numbers.
- (3) The PID control data allocations are shown below.

(a) For incomplete derivative

Specified device number +0	Number of loops		
+1	Number of loops in 1 scan	Conmon to all loops	
+2	Selection of operational expression		
+3	Sampling cycle (Ts)		
+4	Proportional constant (K _P)		
+5	Integral constant (Ti)		
+6	Derivative constant (T _D)		
+7	Filter coefficient (α)		
+8	MV lower limit (MVLL)	For loop No.1	
+9	MV upper limit (MVHL)	(14 words)	
+10	MV change rate limit (△MVL)		
+11	PV change rate limit (△PVL)		
+12	0*		
+13	Derivative gain (K _D)		
+14	0*		
+15	0*	<	
+16	Selection of operational expression		
+17	Sampling cycle (Ts)		
+18	Proportional constant (K _P)		
+19	Integral constant (Ti)		
+20	Derivative constant (T _D)		
+21	Filter coefficient (α)		
+22	MV lower limit (MVLL)	For loop No.2	
+23	MV upper limit (MVHL)	(14 words) For the	
+24	MV change rate limit (\triangle MVL) PV change rate limit (\triangle PVL)	number	
+25 +26	0 *	loops to	be
		useu	
+27 +28	Derivative gain (K _D)		
+20 +29	0*		
+29	1 1		
4-	L-		
to	to		
+(m + 0)	Selection of operational expression		
+(m + 1)	Sampling cycle (Ts)		
+(m + 2)	Proportional constant (K _P)		
+(m + 3)	Integral constant (T ₁)		
+(m + 4)	Derivative constant (T _D)		
+(m + 5)	Filter coefficient (α)		
+(m + 6)	MV lower limit (MVLL)	For loop No.n	
+(m + 7)	MV upper limit (MVHL)	(14 words)	
+(m + 8)	MV change rate limit (△MVL)		
+(m + 9)	PV change rate limit (△PVL)		
+(m + 10)	0*		
+(m + 11)	Derivative gain (K _D)		
+(m + 12)	0*		
+(m + 13)	0 *	J	
	m=(n-	1) × 14+2	

POINT

Store 0 into the " * " marked area of the PID control data. If other than 0 is stored into the " * " marked area, an error occurs and processing is not performed. (Error code: 4100) Spe

(a) Use the following formula to calculate the number of device points to be used when setting the PID control data:

Number of device points = 2 + 14 × n (n: Number of loops to be used)

- (b) Set each data as a binary value.
- (c) If the number of device points for the number of used loops exceeds the last device number of the specified device, an error occurs and processing is not performed. (Error code: 4101)
- (b) For complete derivative

	(
ecified device number +0	Number of loops		
+1	Number of loops in 1 scan	Conmon to all loop	IS
+2	Selection of operational expression]]	
+3	Sampling cycle (Ts)		
+4	Proportional constant (K _P)		
+5	Integral constant (Ti)		
+6	Derivative constant (T _D)		
+7	Filter coefficient (α)	For loop No.1 (10 words)	
+8	MV lower limit (MVLL)	(10 words)	
+9	MV upper limit (MVHL)		
+10	MV change rate limit (△MVL)		
+11	PV change rate limit (\triangle PVL)	J	
+12	Selection of operational expression		
+13	Sampling cycle (Ts)		
+14	Proportional constant (K _P)		
+15	Integral constant (Ti)		
+16	Derivative constant (T _D)		
+17	Filter coefficient (α)	For loop No.2 (10 words)	
+18	MV lower limit (MVLL)	(10 words)	For the total
+19	MV upper limit (MVHL)	}	number of
+20	MV change rate limit (△MVL)		loops to be used
+21	PV change rate limit (△PVL)		useu
+22	Selection of opperational expression		
+23	Sampling cycle (Ts)	For loop No.3	
		(10 words)	
to	to		
+(m + 0)	Selection of opperational expression		
+(m + 1)	Sampling cycle (Ts)		
+(m + 2)	Proportional constant (K _P)		
+(m + 3)	Integral constant (T ₁)		
+(m + 4)	Derivative constant (T _D)		
+(m + 5)	Filter coefficient (α)	For loop No.n (10 words)	
+(m + 6)	MV lower limit (MVLL)	(10 words)	
+(m + 7)	MV upper limit (MVHL)		
+(m + 8)	MV change rate limit (△MVL)		
+(m + 9)	PV change rate limit (△PVL)	JJ	
	m=(n-1)) × 10+2	

 $m=(n-1) \times 10+2$

(a) Use the following formula to calculate the number of device points to be used when setting the PID control data:

Number of device points = $2 + 10 \times n$ (n: Number of loops to be used)

- (b) Set each data as a binary value.
- (c) If the number of device points used for the corresponding loops exceeds the last device number of the specified device, an error occurs and processing is not performed. (Error code: 4101)

- 5.1.1 Number of loops to be used and the number of loops to be executed in a single scan
 - (1) The number of loops to be used means the number of loops for which PID operation is executed. The sampling cycle time is measured for the set number of loops when the PID operation instruction* is executed. PID operation is executed for the loop for which the sampling cycle time reaches or exceeds the set sampling cycle.
 - (2) Processing time increases in proportion to the number of loops for which PID operation is executed when the PID operation instruction is executed.

Processing time = $A + B \times n$

- A : Fixed time for measuring sampling time
- B : Time required to execute PID operation for a loop
- n = Number of loops
- (3) The number of loops to be executed in a single scan means the number of loops for which PID operation is executed in one scan when there is more than one loop for which sampling cycle time reaches or exceeds the set sampling cycle when the PID operation instruction is executed.

If the number of loops to be executed in a single scan is set, PID operation is only executed for the set number of loops even if there are a greater number of loops for which the sampling cycle time reaches or exceeds the set sampling cycle when the PID operation instruction is executed. PID operation is executed for the rest of the loops in the next scan.



POINT

If the number of loops for which sampling cycle time reaches or exceeds the set sampling cycle is greater than the number of loops to be executed in a single scan, the PID operation execution priority is as follows:

- (1) The lowest numbered loop is given the highest priority.
- (2) If there are loops in the preceding scan for which PID operation has not been executed, they are given the highest priority.

REMARK

- *: The following instructions are available as the PID operation instructions.
 - S.PIDCONT (incomplete derivative)
 - PIDCONT (complete derivative)

5.1.2 Sampling cycle

- (1) A sampling cycle is the cycle in which PID operation is executed. The measurement time for one scan is added to the measurement time of up to the preceding scan each time a PID operation instruction*¹ is executed. When the cumulative value reaches or exceeds the set sampling cycle, the PID operation of the corresponding loop is performed.
- (2) The measured value of the sampling time used for PID operation is truncated to units of 10 ms.

For example, if the sampling cycle setting is 50 ms and the measured value is 57 ms, PID operation is executed with a sampling time of 50 ms. If the measured value is 64 ms, PID operation is executed with a sampling time of 60 ms.



The sampling cycle is measured when the PID operation instruction is executed. Therefore, a value smaller than the sequence program scan time cannot be set for the sampling cycle. If a value smaller than the scan time is set, PID operation will be executed in accordance with the scan time.

REMARK

- *1: The following instructions are available as the PID operation instructions.
 - S.PIDCONT (incomplete derivative)
 - PIDCONT (complete derivative)
- *2: The following instructions are available as the PID control data setting instructions.
 - S.PIDINIT (incomplete derivative)
 - PIDINIT (complete derivative)

MEMO

5.2 I/O Data

- (1) The I/O data consists of input data, such as the SV (set value) and PV (process value), which are set to execute PID operation, and output data, such as operation results.
- (2) The I/O data area is divided into the "data area where data are allocated loop-byloop" and "work area used by the system to perform PID operation".

Data Name		Description	Setting Range			
			QC			
			With PID limits	Without PID limits		
Set value	sv	PID control target value	0 to 2000	-32768 to 32767		
Process value	PV	 Feedback data from controlled system objective to A/D converter module 	-50 to 2050	-32768 to 32767		
Automatic manipulated value	MV	 The manipulated value calculated by PID operation. Output from the D/A converter module to the controlled system. 	-50 to 2050	-32768 to 32767		
Process value after filtering	PVf	 Process value calculated using the operation formula in POINT(1) in Section 3.1.2/ POINT(1) in Section 3.2.2. 	-50 to 2050	-32768 to 32767		
Manual manipulated value	MVman	 In the manual control mode, the data output from the D/A converter module is stored. 	-50 to 2050	-32768 to 32767		
Manual/ automatic selection	MAN/ AUTO	 Selects whether the output data to the D/A converter module is shown as a manual manipulated value or an automatic manipulated value. In manual control mode, the automatic manipulated value remains unchanged. 	0: Manual manipulated value 1: Automatic manipulated value	0: Manual manipulated value 1: Automatic manipulated value		
Alarm	ALARM	 Used to determine if the change rate of the MV (manipulated value) and the PV (process value) is within or outside the allowable range. Once set, the alarm data is retained until the user resets it. If the MV is outside the limit range, "1" is set for bit 1 (b1). If the PV is outside the limit range, "1" is set for bit 0 (b0). 	If the PV is outside the limit range, '1' is set for bit 0. If the MV is outside the limit range, '1' is set for bit 1.	If the PV is outside the limit range, '1' is set for bit 0. If the MV is outside the limit range, '1' is set for bit 1.		

Table 5.4 I/O Data List

	QnACPU	Processing when Set Data is Outside the Specified Range	
	0 to 2000	 QCPU "with PID limit", or QnACPU, PID operation is performed after conversion into the following value. When SV is less than 0, SV must be 0. When SV is greater than 2000, SV must be 2000. 	
	-50 to 2050	 QCPU "with PID limit", or QnACPU, PID operation is performed after conversion into the following value. When PV is less than -50, PV must be -50. When PV is greater than 2050, PV must be 2050. 	
	-50 to 2050	_	
	-50 to 2050		
	-50 to 2050	 QCPU "with PID limit", or QnACPU, PID operation is performed after conversion into the following value. When MV_{MAN} is less than -50, MV_{MAN} must be -50. When MV_{MAN} is greater than 2050, MV_{MAN} must be 2050. 	
	0: Manual manipulated value 1: Automatic manipulated value	An error occurs if the setting is neither 0 nor 1, and PID operation of the corresponding loop will not be executed.	
	Lif the PV is outside the limit range, '1' is set for bit 0. If the MV is outside the limit range, '1' is set for bit 1.		

- (3) For the I/O data, any word device number can be specified. However, all the data used for the corresponding loops must be set in devices with consecutive numbers.
- (4) The I/O data allocations are shown below.(a) For incomplete derivative

Designated device number +0	Initial processing flag	} Write		
+1	Work area for PID control	Read/write		
to	(cannot be used)	disabled		
+9	· · ·		-~	<u>`</u>
+10	Set value (SV)			
+11	Process value (PV)			
+12	Automatic manipulated value (MV)	Read		
+13	Process value after filtering (PVf)			
+14	Manual manipulated value (MV _{MAN})	Write	I/O data area	
+15	Manual/automatic selection (MAN/AUTO)		for loop 1	
+16	Alarm (ALARM)	Read/write	(23 words)	
+17	Work area for No. 1 loop	Read/write		
to	(cannot be used)	disabled		
+32 +33	Set value (SV)		-	
+33	Process value (PV)	─		
	Automatic manipulated value (MV)	—		
+35	Process value after filtering (PVf)			
+36 +37	Manual manipulated value (MV _{MAN})	— {		
+37 +38	Manual/automatic selection (MAN/AUTO)	──	I/O data area	
+38 +39	()		for loop 2 (23 words)	
+39	Alarm (ALARM)	Read/write	(23 Words)	Number
to	Work area for No. 2 loop	Read/write		of loops
+55	(cannot be used)	disabled		to be used
+56	Set value (SV)	─- {	-1	
+57	Process value (PV)	Write		
+58	Automatic manipulated value (MV)	Read	I/O data area for loop 3	
+59			(23 words)	
to	to	· · · · · · · · · · · · · · · · · · ·		
+ (m + 0)	Set value (SV)			
+ (m + 1)	Process value (PV)			
+ (m + 2)	Automatic manipulated value (MV)	Read		
+ (m + 3)	Process value after filtering (PVf)			
+ (m + 4)	Manual manipulated value (MV _{MAN})	Write	> I/O data area	
+ (m + 5)	Manual/automatic selection (MAN/AUTO)		for loop n	
+ (m + 6)	Alarm (ALARM)	Read/write	(23 words)	
+ (m + 7)	Work area for No. n loop	Deedhuitte		
to + (m + 22)	(cannot be used)	Read/write disabled		
· (iii · 22)				ر
	$m=(n-1) \times 23$	3+10		

- $m=(n-1) \times 23+10$
- 1) Use the following formula to calculate the number of device points to be used when setting the input/output data:

Number of device points = 10 + 23 × n (n: Number of loops to be used)

2) Set each data as a binary value.

- 3) The initial processing flag sets the processing method at the start of PID operation.
 - •) In the initial PID operation processing cycle, operation is executed assuming that the set sampling cycle is reached or exceeded.
 - •) The initial processing flag is set in the following manner:
 - 0.....PID operation is batch processed in a single scan for the number of loops to be used.
 - Other than 0 PID operation is processed in several scans for the number of loops to be used.
 - Sampling begins sequentially from the loop for which the initial processing has been completed. The number of processing loops per scan is the set
 - number of loops to be executed per scan.
- 4) Where "write" is designated for a data area, it indicates that the data should be written with a user sequence program.
 Where "read" is designated for a data area, it indicates that the data should be read with a user sequence program.
 Never attempt to write data to a data area designated "read/write disabled" or "read". If this is attempted, correct PID operation will not be possible.
- 5) If the number of device points used for the corresponding loops exceeds the last device number of the specified device, an error occurs and processing is not performed. (Error code: 4101)

(b) For complete derivative

+1 0 (cannot be used) Read/write disabled +0 +0 +0 +0 +0 +0 +0 +0 +0 +0 +0 +0 +0 +	Designated device number +0	Initial processing flag]}	Write		
to (cannot be used) (disabled ++9 Set value (SV) Write +11 Process value (PV) Read +12 Automatic manipulated value (MV) Read +13 Process value after filtering (PVf) Read +14 Manual manipulated value (MVANA) Write I/O data area for loop 1 +16 Manual/automatic selection (MAN/AUTO) Read/write I/O data area for loop 1 +17 Work area for No. 1 loop Read/write I/O data area for loop 2 +28 Set value (SV) Write I/O data area for loop 2 +29 Process value (PV) Read I/O data area for loop 2 +30 Automatic manipulated value (MV) Read I/O data area for loop 2 +33 Manual/automatic selection (MAN/AUTO) Read/write I/O data area for loop 2 +34 Alarm (ALARM) Read/write I/O data area for loop 2 +46 Set value (SV) Write I/O data area for loop 3 +46 Set value (PV) Write I/O data area for loop 3 +47 Process value (PV) Write I/O data area for loop 3 +48 Automatic manipulated value (MV) Read I/O data area for loop 3 +49 to I/O data area for	+1	Work area for PID control		Read/write		
+9			}			
+11 Process value (PV) Automatic manipulated value (MV) +12 Automatic manipulated value (MV) +13 Process value after filtering (PVf) +14 Manual/automatic selection (MAN/AUTO) +15 Manual/automatic selection (MAN/AUTO) +16 Alarm (ALARM) +17 Work area for No. 1 loop (cannot be used) +27 Set value (SV) +28 Set value (PV) +30 Automatic manipulated value (MV) +31 Process value after filtering (PVf) +32 Manual/automatic selection (MAN/AUTO) +33 Manual/automatic selection (MAN/AUTO) +34 Alarm (ALARM) +35 Work area for No. 2 loop (cannot be used) +46 Set value (SV) +47 Process value after filtering (PVf) +48 Automatic manipulated value (MV) +48 Automatic manipulated value (MV) +44 Automatic manipulated value (MV) +44 Automatic manipulated value (MV) +47 Process value after filtering (PVf) +48 Automatic manipulated value (MV) +44 Automatic manipulated value (MV) +44 Automatic manipulated value (MV) +47 Process value after filtering (PVf)		,			、	\ \
+11 Process value (PV) +12 Automatic manipulated value (MV) +13 Process value after filtering (PVf) +14 Manual/automatic selection (MAN/AUTO) +16 Alarm (ALARM) +17 Work area for No. 1 loop (cannot be used) Read/write +27 Set value (SV) +28 Set value (SV) +29 Process value after filtering (PVf) +30 Automatic manipulated value (MV) +33 Manual/automatic selection (MAN/AUTO) +34 Manual/automatic selection (MAN/AUTO) +35 Monual/automatic selection (MAN/AUTO) +36 Manual/automatic selection (MAN/AUTO) +46 Set value (SV) +46 Set value (SV) +47 Process value (PV) +48 Automatic manipulated value (MV) +47 Process value (PV) +47 Process value (PV) +48 Automatic manipulated value (MV) +47 Process value (PV) +48 Automatic manipulated value (MV) +47 Process value (PV) +48 Automat			}	Write		
ReadHocess value after filtering (PVf)H13Process value after filtering (PVf)H14Manual manipulated value (MVvms)H15Manual/automatic selection (MAN/AUTO)H16Alarm (ALARM)H17Work area for No. 1 loop (cannot be used)H27Work area for No. 1 loop (cannot be used)H28Set value (SV)H29Process value (PV)H31Process value (PV)H33Process value after filtering (PVf)H34Manual/automatic selection (MAN/AUTO)H35Mork area for No. 2 loop (cannot be used)H46Set value (SV)H47Process value (PV)H48U/O data area for loop 2H44Automatic manipulated value (MV)H44Automatic manipulated value (MV)H44KeadH44Manual/automatic selection (MAN/AUTO)H44Manual manipulated value (MV)H44Manual manipulated value (MV)H45Manual/automatic selection (MAN/AUTO) </td <td></td> <td></td> <td>{</td> <td>White</td> <td></td> <td></td>			{	White		
+13 Process value after filtering (PVf) +14 Manual manipulated value (MVMAN/AUTO) +15 Manual/automatic selection (MAN/AUTO) +16 Alarm (ALARM) +17 Work area for No. 1 loop 10 (cannot be used) +27 Set value (SV) +28 Set value (SV) +29 Process value after filtering (PVf) +30 Automatic manipulated value (MVMAN) +33 Process value after filtering (PVf) +34 Alarm (ALARM) +35 Manual/automatic selection (MAN/AUTO) +46 Set value (SV) +47 Process value (PV) +48 Automatic manipulated value (MV) +44 Automatic manipulated value (MV) +44 Automatic manipulated value (MV) +44 Automatic manipulated value (MV) +47 Process value (PV) +17 Process value (PV) +18 Automatic manipulated value (MV) +48 Automatic manipulated value (MV) +19 Process value after filtering (PVf) +10 Process value after filtering (PVf)	+12	,	}	Read		
+15 Manual/automatic selection (MAN/AUTO) Fread/write for loop 1 +16 Alarm (ALARM) Fread/write for loop 1 +17 Work area for No. 1 loop Fread/write for loop 1 +27 Set value (SV) Write Read/write +29 Process value (PV) Write Read +30 Automatic manipulated value (MV) Read I/O data area +31 Process value after filtering (PVf) Read/write for loop 2 +33 Manual/automatic selection (MAN/AUTO) Read/write for loop 2 +34 Alarm (ALARM) Read/write Read/write for loop 2 +45 Work area for No. 2 loop Read/write for loop 3 to be used +46 Set value (SV) Write V/O data area for loop 3 +47 Process value (PV) Read V/O data area for loop 3 +48 Automatic manipulated value (MV) Read Vrite V/O data area + (m + 0) Set value (SV) Write Vrite V/O data area + (m + 2) Automatic manipulated value (MV) Read <td>+13</td> <td></td> <td>{</td> <td></td> <td></td> <td></td>	+13		{			
+15 Manual/automatic selection (MAN/AUTO) For loop 1 +16 Alarm (ALARM) Read/write (18 words) +17 Work area for No. 1 loop Bead/write (aisabled +27 Set value (SV) Write Read +30 Automatic manipulated value (MV) Read Write +31 Process value after filtering (PVf) Read Write +32 Manual/automatic selection (MAN/AUTO) Read/write I/O data area for loop 2 +33 Manual/automatic selection (MAN/AUTO) Read/write I/O data area for loop 2 +45 Vork area for No. 2 loop Read/write I/O data area for loop 3 +45 Set value (SV) Write I/O data area for loop 3 +46 Set value (SV) Write I/O data area for loop 3 +47 Process value (PV) Write I/O data area for loop 3 +48 Automatic manipulated value (MV) Read I/O data area for loop 3 +49 to Vrite I/O data area for loop 1 + (m + 2) Automatic manipulated value (MV) Read I/O data area for loop 1 + (m + 3)	+14		}	Write		
+177 Work area for No. 1 loop (cannot be used) Read/write disabled +28 Set value (SV) Write +29 Process value (PV) Read +30 Automatic manipulated value (MV) Read +31 Process value after filtering (PVf) Read/write +32 Manual manipulated value (MV.m.n) Write +33 Manual/automatic selection (MAN/AUTO) Read/write +34 Alarm (ALARM) Read/write +45 Work area for No. 2 loop Read/write to Write I/O data area +46 Set value (SV) Write +47 Process value (PV) Write +48 Automatic manipulated value (MV) +49 to +49 to +47 Process value (PV) +(m + 1) Process value after filtering (PVf) +(m + 2) Automatic manipulated value (MV) +(m + 3) Process value after filtering (PVf) +(m + 4) Manual manipulated value (MV/manipulated value (MV/manipul			{			
to +27 +28Cannot be used)disabled+27 +28Set value (SV)Write+29 +29Process value (PV)+30 +31 +32 Manual manipulated value (MV)Read+33 +34 +33 +43Manual/automatic selection (MAN/AUTO)+34 +45 to (cannot be used)Read/write+35 +45 to toWrite+46 +47Set value (SV)+47 +48 toProcess value (PV)+48 +49Automatic manipulated value (MV)+48 to toto+49 toto+41 toProcess value (PV)+42 to to+44 to to+44 to to+45 to to+47 to to+48 to to+49 to+44 to to+47 to to+48 to to to+49 to to+44 to to+44 to to+47 to 	+16	Alarm (ALARM)	{	Read/write	(18 words)	
+27 (Calific Versee) Usabled +28 Set value (SV) Write +29 Process value (PV) +30 Automatic manipulated value (MV) +31 Process value after filtering (PVf) +32 Manual manipulated value (MVMANN) +33 Manual/automatic selection (MAN/AUTO) +34 Alarm (ALARM) +45 Work area for No. 2 loop (cannot be used) Read/write +46 Set value (SV) +47 Process value (PV) +48 Automatic manipulated value (MV) +49 to to to +49 to to Set value (SV) +(m + 0) Set value (SV) +(m + 1) Process value (PV) +(m + 2) Automatic manipulated value (MV) +(m + 2) Automatic manipulated value (MV) +(m + 3) Process value (PV) +(m + 4) Manual manipulated value (MV/MANN) +(m + 5) Manual/automatic selection (MAN/AUTO) +(m + 5) Manual/automatic selection (MAN/AUTO) +(m + 7) Work area for No. n loop to Read/write +(m + 7) Work area for No. n loop to Read/write <td>+17</td> <td>Work area for No. 1 loop</td> <td></td> <td>Read/write</td> <td></td> <td></td>	+17	Work area for No. 1 loop		Read/write		
+28 Set value (SV) +29 Process value (PV) +30 Automatic manipulated value (MV) +31 Process value after filtering (PVf) +32 Manual manipulated value (MV/MANK) +33 Manual manipulated value (MV/MANK) +34 Manual manipulated value (MV/MANK) +35 Manual manipulated value (MV/MANK) +34 Alarm (ALARM) +35 Work area for No. 2 loop (cannot be used) +44 Set value (SV) +47 Process value (PV) +48 Automatic manipulated value (MV) +49 to to to +(m + 0) Set value (SV) +(m + 1) Process value (PV) +(m + 2) Automatic manipulated value (MV) +(m + 3) Process value (PV) +(m + 4) Manual manipulated value (MV) +(m + 4) Manual manipulated value (MV) +(m + 5) Manual manipulated value (MV/MANK) +(m + 6) Alarm (ALARM) +(m + 7) Work area for No. n loop to to to Read/write <td></td> <td>(cannot be used)</td> <td></td> <td>disabled</td> <td></td> <td></td>		(cannot be used)		disabled		
+29 Process value (PV) +30 Automatic manipulated value (MV) +31 Process value after filtering (PVf) +32 Manual manipulated value (MV/MARN) Manual/automatic selection (MAN/AUTO) Read +33 Manual/automatic selection (MAN/AUTO) +34 Alarm (ALARM) *35 Work area for No. 2 loop to Read/write +46 Set value (SV) *47 Process value (PV) +48 Automatic manipulated value (MV) *49 Vinte to to *10 Set value (SV) *11 Process value (PV) *12 Automatic manipulated value (MV) *149 Vinte *10 to *110 Process value (PV) *111 Process value (PV) *112 Automatic manipulated value (MV) *131 Process value (PV) *131 Process value (PV) *131 Process value (PV) *132 Automatic manipulated value (MV) *133 Process value (PV)		Set value (SV)	— {·	4	{	
+30Automatic manipulated value (MV)+31Process value after filtering (PVf)+32Manual manipulated value (MVMAN)+33Manual/automatic selection (MAN/AUTO)+34Alarm (ALARM)+35Work area for No. 2 loop (cannot be used)+46Set value (SV)+47Process value (PV)+48Automatic manipulated value (MV)+49(automatic manipulated value (MV)+49(Beradu and antipulated value (MV)+(m + 0)Set value (SV)+ (m + 1)Process value (PV)+ (m + 2)Automatic manipulated value (MV)+ (m + 3)Process value after filtering (PVf)+ (m + 4)Manual manipulated value (MV/MAN)+ (m + 5)Manual ananipulated value (MV/MAN)+ (m + 6)Alarm (ALARM)+ (m + 7)Work area for No. n loop (cannot be used)+ (m + 7)Work area for No. n loop (cannot be used)+ (m + 7)Work area for No. n loop 			—	Write		
+31Process value after filtering (PVf)Read+32Manual manipulated value (MVMAN)WriteI/O data area for loop 2 (18 words)+33Manual/automatic selection (MAN/AUTO)Read/writeI/O data area for loop 2 (18 words)+34Alarm (ALARM)Read/writeI/O data area for loop 2 (18 words)+45Work area for No. 2 loop (cannot be used)WriteI/O data area for loop 3 (18 words)+46Set value (SV)WriteI/O data area for loop 3 (18 words)+47Process value (PV)ReadI/O data area for loop 3 (18 words)+48Automatic manipulated value (MV)ReadI/O data area for loop 3 (18 words)+49toWriteI/O data area for loop 3 (18 words)+ (m + 0)Set value (SV)WriteRead+ (m + 1)Process value (PV) + (m + 3)WriteRead+ (m + 3)Process value after filtering (PVf) + (m + 4)ReadWrite+ (m + 5)Manual manipulated value (MVMAN) + (m + 7) toWriteI/O data area for loop n (18 words)+ (m + 7)Work area for No. n loop (cannot be used)Read/writeI/O data area for loop n (18 words)			-+			
+32 Manual manipulated value (MV_MAN) Write I/O data area for loop 2 (18 words) +33 Manual/automatic selection (MAN/AUTO) Read/write for loop 2 (18 words) +34 Alarm (ALARM) Read/write for loop 2 (18 words) +45 Work area for No. 2 loop (cannot be used) Read/write I/O data area for loops to be used +46 Set value (SV) Write I/O data area for loop 3 (18 words) I/O data area for loop 3 (18 words) +47 Process value (PV) to to I/O data area for loop 3 (18 words) +49 to to I/O data area for loop 3 (18 words) I/O data area for loop 3 (18 words) + (m + 0) Set value (SV) Write I/O data area for loop 3 (18 words) I/O data area for loop 1 (18 words) + (m + 1) Process value (PV) Read I/O data area for loop 1 (18 words) + (m + 3) Process value after filtering (PVf) Read/write I/O data area for loop 1 (18 words) + (m + 4) Manual/automatic selection (MAN/AUTO) Read/write I/O data area for loop n (18 words) + (m + 7) Work area for No. n loop (cannot be used) Read/write Read/write		· · · · ·		Read		
+33Nonder the formed and the for			—{{			
+34 Alarm (ALARM) Read/write (18 words) +35 Work area for No. 2 loop Read/write Read/write +45 Work area for No. 2 loop disabled I/O data area +46 Set value (SV) Write I/O data area +47 Process value (PV) Read I/O data area +48 Automatic manipulated value (MV) Read I/O data area +49 to to I/O data area + (m + 0) Set value (SV) Write I/O data area + (m + 1) Process value (PV) Read I/O data area + (m + 2) Automatic manipulated value (MV) Read I/O data area + (m + 3) Process value after filtering (PVf) Read Write I/O data area + (m + 5) Manual/automatic selection (MAN/AUTO) Read/write I/O data area for loop n + (m + 7) to Vork area for No. n loop Read/write I/B words) I/B words)				Write	(
Number of loops+35Work area for No. 2 loop (cannot be used)Read/write disabled+45Set value (SV)+47Process value (PV)+48Automatic manipulated value (MV)+49to+49to+41Process value (PV)+42Number of loops to+43to+44Automatic manipulated value (MV)+44Kead+45to+47Process value (PV)+48Number of loop 3 (18 words)+49to+49to+49Vrite+49Vrite+49Vrite+41Process value (PV)+42Number of loop 3 (18 words)+44Number of loop 3 (18 words)+44Vrite+44Number of loop 3 (18 words)+45Y+46Number of loop 3 (18 words)+47Process value (PV) Manual manipulated value (MV)+44Number of loop 1 (18 words)+44Nanual manipulated value (MVMAN)+44Nanual/automatic selection (MAN/AUTO)+44Nanual/automatic selection (MAN/AUTO)+45Nord area for No. n loop (cannot be used)+46Nord area for No. n loop (cannot be used)+47Nord area for No. n loop (cannot be used)+48Nord area for No. n loop (cannot be used)+49Nord area for No. n loop (cannot be used)+49Nord area for No. n loop (cannot b		· · · · · · · · · · · · · · · · · · ·	-+	Pood/write		
to Work area for No. 2 loop Read/write of loops +45 (cannot be used) Write I/O data area +46 Set value (SV) Write I/O data area +47 Process value (PV) Read I/O data area +48 Automatic manipulated value (MV) Read I/O data area +49 to to 18 words) + (m + 0) Set value (SV) Write I/O data area + (m + 1) Process value (PV) Write Read + (m + 2) Automatic manipulated value (MV) Read I/O data area + (m + 3) Process value after filtering (PVf) Read I/O data area + (m + 5) Manual/automatic selection (MAN/AUTO) Read/write I/O data area + (m + 7) Work area for No. n loop Read/write I/O data area + (m + 7) Work area for No. n loop Read/write I/O data area			-			Number
+45 (cannot be used) disabled to be used +46 Set value (SV) Process value (PV) Write I/O data area for loop 3 (18 words) +48 Automatic manipulated value (MV) Read I/O data area for loop 3 (18 words) + (m + 0) Set value (SV) Write Read I/O data area for loop 1 (18 words) + (m + 1) Process value (PV) Write Read I/O data area for loop 1 (18 words) + (m + 2) Automatic manipulated value (MV) Read Write I/O data area for loop 1 (18 words) + (m + 3) Process value after filtering (PVf) Write I/O data area for loop n (18 words) I/O data area for loop n (18 words) + (m + 5) Manual/automatic selection (MAN/AUTO) Read/write I/O data area for loop n (18 words) + (m + 7) Work area for No. n loop (cannot be used) Read/write Read/write		· ·				Sof loops
+47 Process value (PV) +48 Automatic manipulated value (MV) +48 Automatic manipulated value (MV) +49 to to to +(m + 0) Set value (SV) +(m + 1) Process value (PV) +(m + 2) Automatic manipulated value (MV) +(m + 3) Process value after filtering (PVf) +(m + 4) Manual manipulated value (MVMAN) +(m + 5) Manual/automatic selection (MAN/AUTO) +(m + 6) Alarm (ALARM) +(m + 7) Work area for No. n loop to Read/write		(cannot be used)		disabled	J	to be used
+47 Process value (PV) +48 Automatic manipulated value (MV) +49	+46	Set value (SV)	\			
+48 Automatic manipulated value (MV) Read for loop 3 +49 to to (18 words) + (m + 0) Set value (SV) Write Read + (m + 1) Process value (PV) Write Read + (m + 2) Automatic manipulated value (MV) Read I/O data area for loop n + (m + 3) Process value after filtering (PVf) Write I/O data area for loop n + (m + 4) Manual manipulated value (MVMAN) Write I/O data area for loop n + (m + 5) Manual/automatic selection (MAN/AUTO) Read/write I/O data area for loop n + (m + 7) Work area for No. n loop Read/write I/8 words)	+47	Process value (PV)	ſ	vvrite	I/O data area	
toto+ (m + 0)Set value (SV)+ (m + 1)Process value (PV)+ (m + 2)Automatic manipulated value (MV)+ (m + 3)Process value after filtering (PVf)+ (m + 4)Manual manipulated value (MVMAN)+ (m + 4)Manual manipulated value (MVMAN)+ (m + 5)Manual/automatic selection (MAN/AUTO)+ (m + 6)Alarm (ALARM)+ (m + 7)Work area for No. n looptoWork area for No. n loop(cannot be used)Read/write	+48	Automatic manipulated value (MV)	}	Read	/	
+ (m + 0) Set value (SV) + (m + 1) Process value (PV) + (m + 2) Automatic manipulated value (MV) + (m + 3) Process value after filtering (PVf) + (m + 4) Manual manipulated value (MVMAN) + (m + 5) Manual/automatic selection (MAN/AUTO) + (m + 6) Alarm (ALARM) + (m + 7) Work area for No. n loop to Work area for No. n loop	+49		1			
+ (m + 1) Process value (PV) + (m + 2) Automatic manipulated value (MV) + (m + 3) Process value after filtering (PVf) + (m + 4) Manual manipulated value (MVMAN) + (m + 5) Manual manipulated value (MVMAN) + (m + 5) Manual/automatic selection (MAN/AUTO) + (m + 6) Alarm (ALARM) + (m + 7) Work area for No. n loop to Vork area for No. n loop (annot be used) Read/write	to	to			<i>*</i>	
+ (m + 1) Process value (PV) + (m + 2) Automatic manipulated value (MV) + (m + 3) Process value after filtering (PVf) + (m + 4) Manual manipulated value (MVMAN) + (m + 4) Manual manipulated value (MVMAN) + (m + 5) Manual/automatic selection (MAN/AUTO) + (m + 6) Alarm (ALARM) + (m + 7) Work area for No. n loop to Work area for No. n loop to Read/write	+(m+0)	Set value (SV))	
+ (m + 2) Automatic manipulated value (MV) + (m + 3) Process value after filtering (PVf) + (m + 4) Manual manipulated value (MV _{MAN}) + (m + 5) Manual/automatic selection (MAN/AUTO) + (m + 6) Alarm (ALARM) + (m + 7) Work area for No. n loop to (cannot be used)	· · · · · ·			Write		
+ (m + 3) Process value after filtering (PVf) Read + (m + 4) Manual manipulated value (MV _{MAN}) Write + (m + 5) Manual/automatic selection (MAN/AUTO) Write + (m + 6) Alarm (ALARM) Read/write + (m + 7) Work area for No. n loop Read/write			-+			
+ (m + 4) Manual manipulated value (MVMAN) Write I/O data area for loop n + (m + 5) Manual/automatic selection (MAN/AUTO) Read/write for loop n + (m + 6) Alarm (ALARM) Read/write (18 words) + (m + 7) Work area for No. n loop Read/write Read/write	()	,		Read		
+ (m + 5) + (m + 6) + (m + 7) to Write Write Write For loop n (18 words) Read/write Kead/write Kead/write	· · · ·	• • •	—{{			
+ (m + 6) Alarm (ALARM) + (m + 7) to Work area for No. n loop (cannot be used) Read/write (18 words)	()			Write	1	
+ (m + 7) to (cannot be used)	()		-+	Pead/write		
to (cannot be used)			$- \langle$	i teau/ write		
(cannot be used)	· · · · ·			Read/write		
		(cannot be used)			J)
m=(n-1) × 18+10		m=(n-1) × 1	8+10		·	

1) Use the following formula to calculate the number of device points to be used when setting the I/O data:

Number of device points = $10 + 18 \times n$ (n: Number of loops to be used)

2) Set each data as a binary value.

- The initial processing flag sets the processing method at the start of PID operation.
 - •) In the initial PID operation processing cycle, operation is executed assuming that the set sampling cycle is reached or exceeded.
 - •) The initial processing flag is set in the following manner:
 - 0.....PID operation is batch processed in a single scan for the number of loops to be used.
 - Other than 0 PID operation is processed in several scans for the number of loops to be used. Sampling begins sequentially from the loop for which the initial processing has been completed. The number of processing loops per scan is the set number of loops to be executed per scan.
- 4) Where "write" is designated for a data area, it indicates that the data should be written with a user sequence program.
 Where "read" is designated for a data area, it indicates that the data should be read with a user sequence program.
 Never attempt to write data to a data area designated "read/write disabled" or "read". If this is attempted, correct PID operation will not be possible.
 Note that when control is to be started from the initial status, data must
- 5) If the number of device points for the number of used loops exceeds the last device number of the specified device, an error occurs and processing is not performed. (Error code: 4101)

be cleared with a sequence program.

MEMO

6. PID CONTROL INSTRUCTIONS

PID control instructions are defined in the same configuration as High Performance model QCPU, Redundant CPU, Universal model QCPU and QnACPU control instructions.

For details on the configuration of control instructions, see the QCPU (Q mode)/ QnACPU Programming Manual (Common Instructions.)

MEMO

7. HOW TO READ EXPLANATIONS FOR INSTRUCTIONS

The explanations for instructions presented in the next section take the following form.

			plicable CPU	Basic Hig	PLC CPU gh Performance ∆*2		QCPU Proc	cess CPU X	Redunda CPU O	nt QnA	Q4AR ×
8.1.2 PID	Operation									No. are 041 No. are 050	
Set Data	Internal E (System, Bit		e Register	MELSEC	Usable Device NET/10 (H) US 1 Word	Special Fi Module U		Index Reg Zn	ister C	constant	Other
		∽⊢	Command					PIDCON	`	<u>}</u>	
ISET DA	(TA]	Set Data			Descr	intion				Data T	vpe
	ION]	(1) When the	First numbe he S.PIDC D operatior	CONT instr	ruction is e			ampling	cycle is	16-bit b measur	<u> </u>
		(2) With the set valu number	•	ONT instru d process by S or la	iction, PID value (P\ iter, and th	/) in the line operation	O data	a area s sult is sto	et to the	e device	the
		(3) PID ope instructi Section	ion appear	executed in ring first a	n respons fter the se	e to the e t time for	executi sampl	on of the ling cycl	e S.PID e has e	CONT lapsed (see
		If not, P It is not scan. If it is ex	PID contro ion in ever PID operati possible t xecuted m rmal samp	rý scan. ion in a no o execute iore than o	ormal sam the S.PIE	pling cycl CONT in	e will r istructi	not avail ion more	able. than o	nce in o	ne
		If the S executi	kecution ty S.PIDCON ion type pr	pe progra T instructio ogram or	m or low s on has be	speed exe en used i d executio	ecutior n an ir on type	n type pr nterrupt	ogram. program	n, fixed s	can

(2) "O" is appended to those devices that can be used with the instruction.	
The classes of use into which the devices that can be used are divided are as follow	/S.

Device Classification	(evetor	Device n, user)	Register JEE		IET/10 (H) []*3	Special Function	Index Register	Constant*1	Other*1
Classification	Bit	Word	rtegistei	Bit	Word		Zn		
Usable devices ^{*4}	X, Y, M, L, SM, F, B, SB, FX* ² , FY* ²	T* ⁵ , ST* ⁵ , C* ⁵ ,D, W, SD,SW, FD	R, ZR	J[]/X J[]/Y J[]/SB	J[]/W J[]/W	U[]\G[]	Z		P, I, J, U,DX, DY, N, BL, TR, BLY\S

*1 : The devices that can be set are indicated in the "Constant" and "Other" columns.

*2 : FX and FY can only be used with bit data, and FD can only be used with word data. *3 : Can be used with MELSECNET/G,MELSECNET/H and MELSECNET/10.

*4 : For the explanation of the corresponding devices, refer to the QCPU User's Manual (Function Explanation, Programming Fundamentals) or the QnACPU Programming Manual (Basics).

*5 : T, ST and C can be used as the word devices only.

7. HOW TO READ EXPLANATIONS FOR INSTRUCTIONS

MELSEC-Q/QnA



- (6) This shows a CPU module to which instructions are given.
 - \bigcirc : Usable, \triangle : Usable on condition, \times : Unusable
- (7) Indicates the conditions that will cause errors and the error numbers.

MELSEC-Q/QnA

8. INCOMPLETE DERIVATIVE PID CONTROL INSTRUCTIONS AND PROGRAM EXAMPLES

This section explains how to use the PID control instructions for PID control and shows some programming examples.

8.1 PID Control Instructions

8

MELSEC-Q/QnA

		QCPU							
Applicable		PLC CPU		Process CPU	Redundant	QnA	Q4AR		
CPU	Basic	High Performance	Universal	1100033-01-0	CPU				
	Δ*1	∆*2	0	×	0	×	×		

*1: First five digits of serial No. are 04122 or later

*2: First five digits of serial No. are 05032 or later

8.1.1 PID control data settings

		Usable Devices							
Set Data	Internal	Devices	evices		NET/10 (H)	Special Function	Index Degister		
	(Systen	n, User)	File Register	Direct	JE]\E]	Module U[]\G[]	Ũ	Constant	Other
	Bit	Word		Bit	Word	Wodule Or 1/Gr 1	Zn		
S	I		0			—			

Instruction mnemonic	Execution condition	Command	
S.PIDINIT			S.PIDINIT (S)
SP.PIDINIT		Command	SP.PIDINIT S

[SET DATA]

Set Data	Description	Data Type
S	First number of devices in which data for PID control is set	16-bit binary

[FUNCTIONS]

(1) The PID control data for the number of loops to be used, which are set to the device number specified by (s) or later, are entered in the CPU module in a batch, thereby making the PID control possible.
See section 5.1 for details on PID control data

See section 5.1 for details on PID control data

- (2) When the S.PIDINIT instruction is executed at more than one point within a scan, the setting value of the S.PIDINIT instruction closest to the S.PIDCONT instruction is effective.
- (3) The S.PIDINIT instruction must be executed before the S.PIDCONT instruction. PID control is not possible if the S.PIDINIT instruction has not been executed.

[OPERATION ERRORS]

- (1) An operation error will occur, the error flag (SM0) will be turned ON, and an error code will be stored in SD0, in the following cases.
 - When the value set as the PID control data is outside the allowable range.

(Error code: 4100)

- When (MV upper limit) < (MV lower limit). (Error code: 4100)
- When (Number of loops used) < (Number of loops executed in one scan). (Error code:4100)

• When the device range allocated to the PID control data area, designated by (S), exceeds the last device number of the corresponding device.

(Error code: 4101)

• When the "*" area of the PID control data that is mentioned in section 5.1 (3) is not 0. (Error code: 4100)

MELSEC-Q/QnA

		QCPU							
Applicable		PLC CPU		Process CPU	Redundant	QnA	Q4AR		
CPU	Basic	High Performance	Universal	1100033-01-0	CPU				
	∆*1	∆*2	0	×	0	×	×		

*1: First five digits of serial No. are 04122 or later

*2: First five digits of serial No. are 05032 or later

8.1.2 PID operation

		Usable Devices							
Set Data	Internal	Devices		Direct I: :\:		Special Function	Inday Degister		
Sel Dala	(Systen	n, User)	File Register			Special Function	0	Constant	Other
	Bit	Word		Bit	Word	Wodule Or 1/Gr 1	211		
S	-		0			_			

Instruction mnemonic	Execution condition	Command		
S.PIDCONT			S.PIDCONT	<u> </u>
SP.PIDCON		Command	SP.PIDCONT	

[SET DATA]

Set Data	Description	Data Type
S	First number of devices allocated to I/O device area	16-bit binary

[FUNCTION]

- (1) When the S.PIDCONT instruction is executed, the sampling cycle is measured and PID operation is performed.
- (2) With the S.PIDCONT instruction, PID operation is carried out on the basis of the set value (SV) and process value (PV) in the I/O data area set to the device number specified by S or later, and the operation result is stored into the automatically manipulated value (MV) area of the I/O data area.
- (3) PID operation is executed in response to the execution of the S.PIDCONT instruction appearing first after the set time for sampling cycle has elapsed (see Section 5.1.2).
- (4) During PID control, turn ON the control command to execute the S.PIDCONT instruction in every scan.
 If not, PID operation in a normal sampling cycle will not available.
 It is not possible to execute the S.PIDCONT instruction more than once in one scan.
 If it is evenued more than once in one scan.

If it is executed more than once in one scan, PID operation cannot be performed in a normal sampling cycle.

(5) The S.PIDCONT instruction is not available for use in an interrupt program, fixed scan execution type program or low speed execution type program. If the S.PIDCONT instruction has been used in an interrupt program, fixed scan execution type program or low speed execution type program, PID operation cannot be performed in a normal sampling cycle. (6) For (S), designate the first number of the device numbers that are designated in the I/O data area. If file registers (R) are designated for the I/O data area, do not set memory protect ON for the file registers (R).
If memory protect is set ON, correct PID operation will be precluded, although no error will occur.

See Section 5.2 for details on the I/O data area.

(7) Execute the S.PIDCONT instruction in every scan even while the manual manipulated value (MVMAN) is being output in the manual control mode. The bumpless switching cannot be executed if the S.PIDCONT instruction has not been executed.

See Section 4.3.1 for details on the bumpless switching.

(8) Use the READY signal to establish an interlock with respect to the individual modules, so that the S.PIDCONT instruction is executed only when both the A/D converter module for reading the PV (process value) and the D/A converter module for outputting the MV (manipulated value) are normal.*



If the S.PIDCONT instruction is executed while either or both of the modules are faulty, PID operation cannot be executed correctly because the PV (process value) cannot be read correctly and/or the MV (manipulated value) cannot be output correctly.

[OPERATION ERRORS]

- (1) An operation error will occur, the error flag (SM0) will be turned ON, and an error code will be stored in SD0, in the following cases.
 - When the S.PIDINIT instruction is executed before executing the S.PIDCONT instruction. (Error code: 4103)
 - When the value set as the PID control data is outside the allowable range.

(Error code: 4100)

• When the device range allocated to the PID control data area, designated with (S), exceeds the last device number of the corresponding device.

(Error code: 4101)

REMARK

*: For details on the READY signals of the A/D converter module and D/A converter module, refer to the manual for the relevant module.

MELSEC-Q/QnA

			QC	PU			
Applicable	PLC CPU			Process CPU	Redundant	QnA	Q4AR
CPU	Basic	High Performance	Universal	1100633 01 0	CPU		
	∆*1	∆*2	0	×	0	×	×

*1: First five digits of serial No. are 04122 or later

*2: First five digits of serial No. are 05032 or later

8.1.3 Operation stop/start of designated loop no.

	Usable D			Usable Devic	ices					
Set Data		Devices n, User)	File Register	Direct . I		Special Function	Ű	Constant	Other	
	Bit	Word		Bit	Word	Module U[]\G[]	Zn	К, Н		
n	0			_				0	—	

Instruction mnemonic	Execution condition	0	indicates PIDSTOP/PIDRUN
S.PIDSTOP S.PIDRUN		Command	S (n)
SP.PIDSTO SP.PIDRUN		Command	SP. n

[SET DATA]

Set Data	Description	Data Type
n	Loop number at which start/stop is to be executed	16-bit binary

[FUNCTION]

- (1) S.PIDSTOP, SP.PIDSTOP
 - (a) Stops the PID operation for the loop number designated by n. The loop stopped by the S.PIDSTOP instruction does not resume PID operation even if the S.PIDINIT instruction is executed.
 - (b) Retains the operation data during the stop.
- (2) S.PIDRUN, SP.PIDRUN
 - (a) Starts the PID operation of the loop No. specified by (n).
 - This instruction is designed to re-execute PID operation of the loop No. that has stopped with the S.PIDSTOP instruction.
 - (b) This instruction will be ignored if the instruction is executed for the loop No. that is currently running PID operation.

[OPERATION ERRORS]

- (1) An operation error will occur and the error flag (SM0) will be turned ON in the following cases.
 - When the loop number designated by (n) does not exist. (Error code: 4100)
 - When (n) is outside the range 1 to 8. (Basic model QCPU) (Error code: 4100)
 - When (n) is outside the range 1 to 32. (High Performance model QCPU, Redundant CPU, Universal model QCPU) (Error code: 4100)
 - When the S.PIDINIT and S.PIDCONT instructions have not been executed before execution of the S.PIDSTOP instruction. (Error code: 4103)
 - When the S.PIDINIT and S.PIDCONT instructions have not been executed before execution of the S.PIDRUN instruction. (Error code: 4103)

MELSEC-Q/QnA

			PU				
Applicable	PLC CPU			Process CPU	Redundant	QnA	Q4AR
CPU	Basic	High Performance	Universal	1100633 01 0	CPU		
	∆*1	∆*2	0	×	0	×	×

*1: First five digits of serial No. are 04122 or later

*2: First five digits of serial No. are 05032 or later

8.1.4 Parameter change at designated loop

					Usable Devid				
Set Data	Internal (System		File Register		NET/10 (H) J[]\[]	Special Function	Ũ	Constant	Other
	Bit	Word		Bit	Word	Module Or 1/Gr 1	211	К, Н	
n	0	0		_			0	_	
S	—		0			—			—



[SET DATA]

Set Data	Description	Data Type
n	Loop number for which change is to be made	
S	First number of devices in which PID control data to be	16-bit binary
•	changed is stored	

[FUNCTIONS]

- (1) Changes the operation parameter for the loop number designated by (n) to the PID control data stored in the devices starting with the device number designated by (s).
- (2) The configuration of the data for PID control which starts from the device number designated by (s) is shown below. For details on PID control data, see Section 5.1.

(S) +0	Selection of operational expression
(S) +1	Sampling cycle (Ts)
(S) +2	Proportional constant (K _P)
(S) +3	Integral constant (Ti)
(S) +4	Derivative constant (T _D)
(S) +5	Filter coefficient (α)
(S) +6	MV lower limit (MVLL)
(S) +7	MV upper limit (MVHL)
(S) +8	MV change rate limit (△MVL)
(S) +9	PV change rate limit (△PVL)
(S) +10	0
(S) +11	Derivative gain (K _D)
(S) +12	0
(S) +13	0

[OPERATION ERRORS]

- (1) An operation error will occur and the error flag (SM0) will be turned ON, and error code will be stored in SD0, in the following cases.
 - When the loop number designated by (n) does not exist. (Error code: 4100)
 - When (n) is outside the range 1 to 8.(Basic model QCPU) (Error code: 4100)
 - When (n) is outside the range 1 to 32.(High Performance model QCPU, Redundant CPU, Universal model QCPU) (Error code: 4100)
 - When the PID control data is outside the setting range. (Error code: 4100)
 - When any of S+10, S+12 and S+13 in the PID control data is not 0.

(Error code: 4100)

- When the device range assigned to the PID control data area by (S) exceeds the last device number of the applicable range. (Error code: 4101)
- When the S.PIDINIT instruction has not been executed before execution of the S.PIDPRMW instruction. (Error code: 4100)

8.2 PID Control Program Examples

This section describes examples of sequence programs that execute PID control.

8.2.1 System configuration for program examples

The following illustrates the system configuration for the program examples in Sections 8.2.2 and 8.2.3



Q64AD I/O numbersX/Y80 to X/Y8F Q62DA I/O numbersX/YA0 to X/YAF

8.2.2 Program example for automatic mode PID control

This section gives a program example in which PID operation is performed using the digital values imported from the Q64AD as PV and the MV obtained as a result of PID operation are output from the Q62DA to control external devices.

[PROGRAMMING CONDITIONS]

- (1) Refer to Section 8.2.1 for details on the system configuration.
- (2) PID operation is executed for 2 loops.
- (3) The sampling cycle is 1 second.

- (6) The following SV are set for loop 1 and loop 2 using a sequence program:

Loop 1	600
Loop 2	1000

- (7) The following devices are used for PID control start/stop commands.
 PID control start commandX0
 PID control stop commandX1
- (8) The digital values of the Q64AD and Q62DA are set within the range 0 to 2000.

REMARK

- *1: For details on PID control data, see Section 5.1.
- *2: For details on I/O data, see Section 5.2.

MELSEC-Q/QnA

[PROGRAM EXAMPLE]

[MOV K2 D501 Sets the number of loops to be executed per scan to 2. common data of PID control data [MOV K0 D502 Sets the operation expression to forward operation. Sets the sampling cycle to 1s. [MOV K100 D503 Sets the proportional constant to 1. Sets the proportional constant to 1.	
[MOV K100 D503] Sets the sampling cycle to 1s.	
[MOV K30000 D505] Sets the integral constant to 3000s.	
[MOV K0 D506] Sets the derivative constant to 0s.	
[MOV K0 D507] Sets the filter coefficient to 0%.	
[MOV K0 D508] Sets the MV lower limit to 0. Setting of F	PID
[M0V K2000 D509] Sets the MV upper limit to 2000.	
[M0V K2000 D510] Sets the MV change rate limit to 2000.	
[MOV K2000 D511] Sets the PV change rate limit to 2000.	
[MOV K0 D512] Sets 0.	
[MOV K800 D513] Sets the derivative gain to 8.	
[MOV K0 D514] Sets 0.	
[MOV K0 D515] Sets 0.	

MELSEC-Q/QnA

SMA22 UNV K1 D516 Jests the operation operation. 1 (M0V K100 D517 Jests the amping oycle to 1s. 1 (M0V K100 D517 Jests the amping oycle to 1s. 1 (M0V K100 D517 Jests the ampontonal constant to 1. 1 (M0V K30000 D519 Jests the direvative constant to 3000s. 1 (M0V K0 D520 Jests the direvative constant to 3000s. 1 (M0V K0 D521 Jests the direvative constant to 3000s. 1 (M0V K0 D521 Jests the filter coeffic ent to 0%. 1 (M0V K2000 D523 Jests the MV change rate limit to 2000. 1 (M0V K2000 D525 Jests the drivative gain to 8. 1 (M0V K00 D527 Jests the drivative gain to 8. 1 (M0V K00 D528 Jests the drivative gain to 8. 1 (M0V K00 D529 Jests to. 1 (M0V	ta
Likov Kitoo Likov Kitoo Likov Kitoo Likov Likov <td< td=""><td>ta</td></td<>	ta
Image: Intervention Constant to 1. Image: Ima	ta
Image: Constant to 3000s. Constant to 3000s. Image: Constant to 3000s. Mov Image: Constant to 3000s. Sets the derivative constant to 300s. Image: Constant to 3000s. Sets the derivative constant to 300s. Image: Constant to 3000s. Sets the derivative constant to 300s. Image: Constant to 3000s. Sets the derivative constant to 300s. Image: Constant to 3000s. Sets the derivative constant to 300s. Image: Constant to 3000s. Sets the MV lower limit to 3000s. Image: Constant to 3000s. Sets the MV lower limit to 00. Image: Constant to 3000s. Sets the MV lower limit to 3000s. Image: Constant to 3000s. Sets the MV lower limit to 2000. Image: Constant to 3000s. Sets the MV change rate limit to 2000. Image: Constant to 3000s. Sets the MV change rate limit to 2000. Image: Constant to 3000s. Sets the derivative gain to 8. Image: Constant to 3000s. Sets the derivative gain to 8. Image: Constant to 3000s. Sets 0. Image	ta
Image: Control of the second	ta
Imov No DS1 J ent to 0%. Imov K0 D522 J Sets the MV lower limit to 0. Sets the MV lower limit to 0. Sets the MV upper limit to 2000. Sets the MV change rate limit to 2000. Sets the PV change rate limit to 2000. Sets the PV change rate limit to 2000. Sets the PV change rate limit to 2000. Sets the derivative gain to 8. Imov K2000 D524 J Sets the derivative gain to 8. Sets the derivative gain to 8. Imov K000 D527 J Sets the PID control data that are set in D500 to D529. Sets the PID control data that are set in D500 to D529. Sets the PID control data that are set in D500 to D529. Sets the PID control data that are set in D500 to D529.	ta
Imov KO b522 b0. Sets the MV upper limit to 2000. Imov K2000 b523 b5 sets the MV change rate limit to 2000. Imov K2000 b524 b5 sets the MV change rate limit to 2000. Imov K2000 b525 b5 sets the MV change rate limit to 2000. Imov K2000 b525 b5 sets the PV change rate limit to 2000. Imov K2000 b526 b5 sets the PV change rate limit to 2000. Imov K00 b526 b5 sets the derivative gain to 8. Imov K800 b527 b5 sets the derivative gain to 8. Imov K00 b528 b5 sets 0. Imov K0 b528 b5 sets 0. Imov K0 b529 b5 sets 0. SM402 [S. PIDINIT b500 b529. Sets the PID control data that are set in b500 to b529. b529. b510 to b529.	ta
Image: Mov K2000 D523 D528 D528 D5200. Image: Mov K2000 D524 D524 D528 D529	
Image:	
Imov K2000 6020 Image: rate limit to 2000. Imov K0 D526 Image: Sets 0. Imov K800 D527 Image: Sets the derivative gain to 8. Imov K0 D528 Image: Sets 0. Imov K0 D528 Image: Sets 0. Imov K0 D529 Image: Sets 0. Imov K0 D529 Image: Sets 0. Imov K0 D529 Image: Sets 0. Image: Sets the PID control data that are set Image: Sets the PID control data that are set Image: Sets the PID control data that are set Image: Set 0. Image: Set 0.	
Image: Set in the derivative gain to 8. Set in th	
Image:	
62 [MOV K0 D529] Sets 0. 62 [S. PIDINIT D500] Sets the PID control data that are set in D500 to D529. X0A0 U0A	
62 SM402 62 [S. PIDINIT 0 Sets the PID control data that are set in D500 to D529. X0A0 U0A	
62 [S. PIDINIT D500] Sets the PID control data that are set in D500 to D529.	
[SET Y0A9] Sets the Q62DA to output enable.	
76 X0A0 X0A9 Y0A9 [RST Y0A9]	
80 [M0V K0 D600] Sets the initial processing flag to 0. of I/O data	data
[M0V K600 D610] Sets the SV to 600.	
[M0V K0 D615] Sets the automatic data for lo mode.	op 1
[M0V K1000 D633] Sets the SV to 1000.	I/O
[MOV K0 D638] Sets the automatic mode.	
91 SET M0 PID operation start command	ioh z

MELSEC-Q/QnA



8.2.3 Program example for changing the PID control mode between automatic and manual

An example program for switching between automatic and manual modes while executing PID operation is described below.

[PROGRAMMINGCONDITIONS]

- (1) Refer to Section 8.2.1 for details on the system configuration.
- (2) PID operation is executed for 1 loop.
- (3) The sampling cycle is 1 second.

(5) The I/O data is set in the following devices:	
Common data	D600 to D609
Loop 1 data	D610 to D630

- (6) The SV and MV in manual mode are set with external digital switches as follows: SV......X30 to X3F MV (manual control mode).....X20 to X2F

(OFF: Automatic mode, ON: Manual mode)

(8) The digital values of the Q64AD and Q62DA are set within the range 0 to 2000.

(9) The PID bumpless processing flag, SM794, is set to OFF. In the manual mode, the SV is automatically rewritten to the PV when PID operation is performed. Therefore, when the manual mode is returned to the automatic mode, the SV must be rewritten to the one used in the automatic mode before switching to the manual mode.

The SV is rewritten step-by-step 10 times as illustrated below:



The SV is rewritten using the operation method illustrated below:



The incremental value obtained with the formula above is added to SV every second. The remainder is added in the first addition operation.

MELSEC-Q/QnA

[PROGRAM EXAMPLE]



MELSEC-Q/QnA



This chapter explains the PID control instruction usage and program examples for implementing PID control.

9.1 PID Control Instructions

MELSEC-Q/QnA

		QCPU						
Applicable		PLC CPU			Redundant	QnA	Q4AR	
CPU	Basic	High Performance	Universal	Process CPU	CPU			
	∆*1	0	0	×	0	0	0	

*1: First five digits of serial No. are 04122 or later

9.1.1 PID control data settings

	Usable Devices								
Set Data	Internal Devices (System, User)		File Register	MELSECNET/10 (H) Direct J[]\[]		Special Function	0	Constant	Other
	Bit	Word		Bit	Word	Module U[]\G[]	Zn		
S	_		0			—			

Instruction mnemonic	Execution condition	Command		
PIDINIT			 PIDINIT	<u> </u>
PIDINITP		Command	PIDINITP	<u> </u>

[SET DATA]

Set Data	Description	Data Type
S	First number of devices in which data for PID control is set	16-bit binary

[FUNCTIONS]

(1) The PID control data for the number of loops to be used, which are set to the device number specified by (s) or later, are entered in the CPU module in a batch, thereby making the PID control possible.

Refer to Section 5.1 for details of the PID control data.

- (2) When the PIDINIT instruction is executed at more than one point within a scan, the setting value of the PIDINIT instruction closest to the PIDCONT instruction is effective.
- (3) The PIDINIT instruction must be executed before the PIDCONT instruction. PID control is not possible if the PIDINIT instruction has not been executed.

[OPERATION ERRORS]

- (1) An operation error will occur, the error flag (SM0) will be turned ON, and an error code will be stored in SD0, in the following cases.
 - When the value set as the PID control data is outside the allowable range.

(Error code: 4100)

- When (Number of loops used) < (Number of loops executed in one scan). (Error code: 4100)
- When (MV upper limit value) < (MV lower limit value).

(Error code: 4100)

• When the device range allocated to the PID control data area, designated by (S), exceeds the last device number of the corresponding device.

(Error code: 4101)

MELSEC-Q/QnA

	QCPU						
Applicable		PLC CPU			Redundant	QnA	Q4AR
CPU	Basic	High Performance	Universal	Process CPU	CPU		
	∆*1	Δ*1 Ο Ο			0	0	0

*1: First five digits of serial No. are 04122 or later

9.1.2 PID control

	Usable Devices								
Set Data	Internal Devices (System, User) File Register		File Register	Direct J. 3.		Special Function	0	Constant	Other
	Bit	Word		Bit	Word	Module UI 3\GI 3	Zn		
S	_		0			_			

Instruction mnemonic	Execution condition	Control command
PIDCONT		Control command PIDCONT S
PIDCONTP		Control command PIDCONTP S

[SET DATA]

Set Data	Description	Data Type
S	First number of devices allocated to I/O device area	16-bit binary

[FUNCTION]

- (1) When the PIDCONT instruction is executed, the sampling cycle is measured and PID operation is performed.
- (2) With the PIDCONT instruction, PID operation is carried out on the basis of the set value (SV) and process value (PV) in the I/O data area set to the device number specified by S or later, and the operation result is stored into the automatically manipulated value (MV) area of the I/O data area.
- (3) PID operation is executed in response to the execution of the PIDCONT instruction appearing first after the set time for sampling cycle has elapsed (see Section 5.1.2).
- (4) During PID control, turn ON the control command to execute the PIDCONT instruction in every scan.
 - If not, PID operation in a normal sampling cycle will not available. It is not possible to execute the PIDCONT instruction more than once in one scan.

If it is executed more than once in one scan, PID operation cannot be performed in a normal sampling cycle.

(5) The PIDCONT instruction is not available for use in an interrupt program, fixed scan execution type program or low speed execution type program. If the PIDCONT instruction has been used in an interrupt program, fixed scan execution type program or low speed execution type program, PID operation cannot be performed in a normal sampling cycle.

- (6) For (S), designate the first number of the device numbers that are designated as the I/O data area. If file registers (R) are designated for the I/O data area, do not set memory protect ON for the file registers (R).
 If memory protect is set ON, correct PID operation will be precluded, although no error will occur.
 See Section 5.2 for details on the I/O data area.
- (7) Execute the PIDCONT instruction in every scan even while the manual manipulated value (MV_{MAN}) is being output in the manual control mode. The bumpless function cannot be executed if the PIDCONT instruction has not been executed.

See Section 4.3.1 for details on the bumpless function.

(8) Use the READY signal to establish an interlock with respect to the individual modules, so that the PIDCONT instruction is executed only when both the A/D converter module for reading the PV (process value) and the D/A converter module for outputting the MV (manipulated value) are normal.*



If the PIDCONT instruction is executed while either or both of the modules are faulty, PID operation cannot be executed correctly because the PV (process value) cannot be read correctly and/or the MV (manipulated value) cannot be output correctly.

[OPERATION ERRORS]

- (1) An operation error will occur, the error flag (SM0) will be turned ON, and an error code will be stored in SD0, in the following cases.
 - When the PIDINIT instruction is executed before executing the PIDCONT instruction. (Error code: 4103)
 - When the value set as the PID control data is outside the allowable range.

(Error code: 4100)

• When the device range allocated to the PID control data area, designated with (S), exceeds the last device number of the corresponding device.

(Error code: 4101)

REMARK

*: For details on the READY signals of the A/D converter module and D/A converter module, refer to the manual for the relevant module.

MELSEC-Q/QnA

	QCPU						
Applicable		PLC CPU			Redundant	QnA	Q4AR
CPU	Basic	High Performance	Universal	Process CPU	CPU		
	×	×	×	×	×	0	0

9.1.3 Monitoring PID control status (QnACPU only)

Set Data	Internal (Systen	Devices n, User)	File Register		JET/10 (H) J[]\[]	Special Function	Ũ	Constant	Other
	Bit	Word		Bit	Word	Wodule Or 1/Gr 1	Zn	К, Н	
n		0				0	—		
S 1		0					0	—	
\$2		0						—	—

Instruction mnemonic	Execution condition	Command	
PID57			PID57 (n) (S) (S2)
PID57P		Command	PID57P (n) (S) (S2

[SET DATA]

Set Data	Description	Data Type				
n	First I/O number of the AD57(S1) used to monitor the PID					
	control status					
6)	Screen number corresponding to the loop number to be	16-bit binary				
0	monitored					
62	Initial screen display request					

[FUNCTION]

- The display unit of the AD57(S1) designated by n displays the PID control status of the loop number designated by n in a bar graph.
 By executing the initial screen display request, designated by n, the characters in the still portion of the monitor screen (with the exception of bar graphs and numerical data) are displayed in the initial state of PID control monitoring.
- (2) Addresses 0 to 1599 in the VRAM area of the AD57(S1) are used for the PID control monitor.
 Therefore, these addresses cannot be used by the user if DID control atotus.

Therefore, these addresses cannot be used by the user if PID control status monitoring is executed; if they are, the data stored in them will be lost.

(3) Execute the CMODE instruction (AD57 command) to monitor the PID control status before executing the PID57 instruction.If the CRT standard display mode, set with the CMODE instruction, has not been set for the AD57(S1), the display unit will not be able to display anything.

(4) Execute the PID57 instruction only after the PIDINIT and PIDCONT instructions have been executed.

An error will occur if the PID57 instruction is executed before the PIDINIT and PIDCONT instructions.

(5) Designate the loop number indicated by (5) with a screen number from "1" to "4", as shown below:

Screen Number	Loop Numbers to be Monitored
1	Loop 1 to loop 8
2	Loop 9 to loop 16
3	Loop 17 to loop 24
4	Loop 25 to loop 32

(6) The initial screen display request, designated by ②, displays the characters in the still portion of the monitor screen.

To make the initial screen display request, set "0" for S2.

Characters besides the bar graphs and numeric data will be not displayed unless the initial screen display request is executed.

- (7) After the initial screen is displayed, the value designated by (5) is automatically stored in (5) and then the PID control monitor function is executed.
 If the device designated by (5) is a file register, do not set the memory protect function for the file register ON.
 If the memory protect function is ON, the screen cannot display the monitor data
- (8) The initial screen display request should only be executed once in response to the first PID57 instruction after the start of QnACPU operation. If it is executed every scan, the bar graphs and numeric data will not be displayed, although the characters in the still portion are displayed.
- (9) To monitor PID control status with the AD57(S1), a character generator ROM and canvas ROM must be loaded to the AD57(S1).
 The characters shown in Figure 9.1, corresponding to character codes 000 to 00BH, must be created in the character generator ROM.
 If these characters are not created, bar graphs cannot be displayed.
 Refer to the following manuals for details on creating the character generator ROM and canvas ROM.

SW1GP-AD57P Operating Manual

correctly.

MELSEC-Q/QnA



Fig. 9.1 Characters for PID Control Status Monitor

[OPERATION ERRORS]

- (1) An operation error will occur and the error flag (SM0) will be turned ON, and an error code will be stored in SD0, in the following cases.
 - When the CMODE instruction has not been executed for AD57(S1).

(Error code: 2110)

- When the PIDINIT instruction has not been executed before the PID57 instruction. (Error code: 4103)
- When the PIDCONT instruction has not been executed before the PID57 instruction. (Error code: 4103)
- When the screen number designated with (5) is outside the range of 1 to 4.

(Error code: 4100)

MELSEC-Q/QnA

	QCPU						
Applicable	PLC CPU			Process CPU	Redundant	QnA	Q4AR
CPU	Basic	High Performance	Universal	1100033 01 0	CPU		
	Δ*1	0	0	×	0	0	0

*1: First five digits of serial No. are 04122 or later

9.1.4 Operation stop/start of designated loop no.

				Usable Devic	ces			
Set Data	Internal Devices		MELSEC	NET/10 (H)	Special Function	Index Degister	Constant	
Sei Dala	(System, User)	File Registe	er Direct	t J[]\[]	Module U[]\G[]	Index Register Zn	Constant K, H	Other
	Bit Word		Bit	Word		20	ΙΧ, Π	
n				0				—
	mnemonic ca PIDSTOP PIDRUN PIDSTOPP PIDRUNP		Command 		[® PIDSTOP/	PIDRUN
[SET D	DATAJ							
		Set Data			scription		Data T	
		n	Loop number at	which start/s	stop is to be exec	uted	16-bit binary	
[FUNC								
 (1) PIDSTOP, PIDSTOPP (a) Stops the PID operation for the loop number designated by The loop stopped by the PIDSTOP instruction does not resume PID operation even if the PIDINIT instruction is executed. (b) Retains the operation data during the stop. (2) PIDRUN, PIDRUNP (a) Starts the PID operation of the loop No. specified by This instruction is designed to re-execute PID operation of the loop No. that has stopped with the PIDSTOP instruction. (b) This instruction will be ignored if the instruction is executed for the loop No. that is currently running PID operation. 					No. that			
[OPER	ATION ERR	ORS]						
 (1) An operation error will occur and the error flag (SM0) will be turned ON in the following cases. When the loop number designated by n does not exist. (Error code: 4100) When n is outside the range 1 to 8. (Basic model QCPU) (Error code: 4100) When n is outside the range 1 to 32.(High Performance model QCPU, Redundant CPU, Universal model QCPU, QnACPU) (Error code: 4100) When the PIDINIT and PIDCONT instructions have not been executed before execution of the PIDSTOP instruction. (Error code: 4103) When the PIDINIT and PIDCONT instructions have not been executed before execution of the PIDRUN instruction. (Error code: 4103) 								

MELSEC-Q/QnA

*1: First five digits of serial No. are 04122 or later

	QCPU						
Applicable		PLC CPU			Redundant	QnA	Q4AR
CPU	Basic	High Performance	Universal	Process CPU	CPU		
	∆*1	0	0	×	0	0	0

9.1.5 Parameter change at designated loop

	Usable Devices								
Set Data	Internal (Systen	Devices n, User)	File Register		NET/10 (H) J[]\[]	Special Function	0	Constant	Other
	Bit	Word		Bit	Word	INIOQUIE O. 1/G. 1	Zn	К, Н	
n	0		0	0 0			—		
S	_		0			_			_



[SET DATA]

Set Data	Description	Data Type
n	Loop number for which change is to be made	
S	First number of devices in which PID control data to be	16-bit binary
	changed is stored	

[FUNCTIONS]

- (1) Changes the operation parameter for the loop number designated by (n) to the PID control data stored in the devices starting with the device number designated by (s).
- (2) The configuration of the data for PID control which starts from the device number designated by (s) is shown below. For details on PID control data, see Section 5.1.
 - S +0 Selection of operation expression
 - (S) +1
 Sampling cycle (Ts)

 (S) +2
 Proportional constant (KP)
 - S +2
 Proportional constant (KP)

 S +3
 Integral constant (Ti)
 - (S) +3 Integral constant (T) (S) +4 Derivative constant (T_D)
 - (s) +5 Filter coefficient (α)
 - (S) +6 MV lower limit (MVLL)
 - (\$) +7 MV upper limit (MVHL)
 - (S) +8 MV change rate limit (Δ MVL)
 - \$ +9 PV change rate limit (\triangle PVL)

[OPERATION ERRORS]

- (1) An operation error will occur and the error flag (SM0) will be turned ON, and error code will be stored in SD0, in the following cases.
 - When the loop number designated by (n) does not exist. (Error code: 4100)
 - When (n) is outside the range 1 to 8. (Basic model QCPU) (Error code: 4100)
- When (n) is outside the range 1 to 32. (High Performance model QCPU, Redundant CPU, Universal model QCPU, QnACPU) (Error code: 4100)
 - When the PID control data is outside the setting range. (Error code: 4100)
 - When the device range assigned to the PID control data area by (S) exceeds the last device number of the applicable range. (Error code: 4101)
 - When the PIDINIT instruction has not been executed before execution of the PIDPRMW instruction. (Error code: 4103)

MELSEC-Q/QnA

9.2 PID Control Program Examples (QCPU only)

This section describes examples of sequence programs that execute PID control.

9.2.1 System configuration for program examples

The following illustrates the system configuration for the program examples in Sections 9.2.2 and 9.2.3.



Q64AD I/O numbersX/Y80 to X/Y8F Q62DA I/O numbersX/YA0 to X/YAF

9.2.2 Program example for automatic mode PID control

This section gives a program example in which PID operation is performed using the digital values imported from the Q64AD as PV and the MV obtained as a result of PID operation are output from the Q62DA to control external devices.

[PROGRAMMING CONDITIONS]

- (1) Refer to Section 9.2.1 for details on the system configuration.
- (2) PID operation is executed for 2 loops.
- (3) The sampling cycle is 1 second.

- (6) The following SV are set for loop 1 and loop 2 using a sequence program:

Loop 1	600
Loop 2	1000

- (7) The following devices are used for PID control start/stop commands.
- (8) The digital values of the Q64AD and Q62DA are set within the range 0 to 2000.

REMARK

- *1: For details on PID control data, see Section 5.1.
- *2: For details on I/O data, see Section 5.2.

MELSEC-Q/QnA

[PROGRAM EXAMPLE]



SM402 Setting of common data Sets the initial 59 ┥┝ -[MOV K0 D600 processing flag to 0. of I/O data -[mov K600 D610 Sets the SV to 600. Setting of I/O data for loop 1 Sets the automatic -FMOV K0 D615 mode. -[mov K1000 D628 Sets the SV to 1000. Setting of I/O data for loop 2 Sets the automatic -[MOV K0 D633 mode. X0 -SET 70 MO +PID operation start command X80 181 MO Sets the PV from the Q64AD to the I/O data area (for loop 1). -FMOV D611 G11 72 ┥┟ 18/ Sets the PV from the Q64AD to the I/O data -[MOV G12 D629 area (for loop 2). MO PIDCONT D600 82 ┥┟ PID operation XOAO MO -[SET Turns ON the output enable of CH. 1 of the Q62DA. Y0A1 85 ł -[SET YOA2 Turns ON the output enable of CH. 2 of the Q62DA. UOA\ -[mov D612 Writes the MV of loop 1 to CH. 1 of G1 the Q62DA. UOA\ -FMOV D630 Writes the MV of loop 2 to CH. 2 of G2 the Q62DA. X1 ┨┠ 97 -[RST MO PID operation stop

MELSEC-Q/QnA

9.2.3 Program example for changing the PID control mode between automatic and manual

An example program for switching between automatic and manual modes while executing PID operation is described below.

[PROGRAMMINGCONDITIONS]

- (1) Refer to Section 9.2.1 for details on the system configuration.
- (2) PID operation is executed for 1 loop.
- (3) The sampling cycle is 1 second.

(4) The PID control data is set in the following	g devices:
Common data	D500 and D501
Loop 1 data	D502 to D511

(5) The I/O data is set in the following devices:	
Common data	D600 to D609
Loop 1 data	D610 to D627

(6) The SV and MV in manual mode are set with external digital switches as follows:
 SV......X30 to X3F
 MV (manual control mode).....X20 to X2F

(7) The following devices are used to start and sto automatic/manual changeover command:	op PID control and the
PID control start command	X0
PID control stop command	X1
SV setting command	X3
MV setting command in manual mode	X4
Automatic/manual mode changeover commar	nd X6

(OFF: Automatic mode, ON: Manual mode)

(8) The digital values of the Q64AD and Q62DA are set within the range 0 to 2000.

(9) The PID bumpless processing flag, SM774, is set to OFF. In the manual mode, the SV is automatically rewritten to the PV when PID operation is performed. Therefore, when the manual mode is returned to the automatic mode, the SV must be rewritten to the one used in the automatic mode before switching to the manual mode.



The SV is rewritten using the operation method illustrated below:



The incremental value obtained with the formula above is added to SV every second. The remainder is added in the first addition operation.




9.3 PID Control Program Examples (QnACPU only)

This section describes examples of sequence programs that execute PID control.

9.3.1 System configuration for program examples

The following illustrates the system configuration for the program examples in Sections 9.3.2 and 9.3.3.



9.3.2 Program example for automatic mode PID control

This section gives a program example in which PID operation is performed using the digital values imported from the A68AD as PV and the MV obtained as a result of PID operation are output from the A62DA to control external devices.

[PROGRAMMING CONDITIONS]

- (1) Refer to Section 8.3.1 for details on the system configuration.
- (2) PID operation is executed for 2 loops.
- (3) The sampling cycle is 1 second.

- (6) The following SV are set for loop 1 and loop 2 using a sequence program:

Loop 1	600
Loop 2	1000

(7) The following devices are used for PID control start/stop commands and the monitoring command with AD57.

(8) The digital values of the A68AD and A62DA are set within the range 0 to 2000.

REMARK

- *1: For details on PID control data, see Section 5.1.
- *2: For details on I/O data, see Section 5.2.

0	SM402		—[MOV	K2	D500	н	Sets the number of loops to be used to "2"	Setting of common
			[MOV	K2	D501	Ъ	Sets the number of PID operation execution loops per scan to "2"	data of PID control data
			—[MOV	KO	D502	Ъ	Sets the operation expression to forward operation	
			——[MOV	K100	D503	Ъ	Sets the sampling cycle to "1 s"	
			—E MOV	K100	D504	Ъ	Sets the proportional constant to "1"	
		[MOV	K30000		D505	н	Sets the integral constant to "3000 s"	
			[MOV	KO	D506	н	Sets the derivative constant to "0 s"	Setting of PID control data
			— [MOV	KO	D507	Н	Sets the filter coefficient to "0 %"	for loop 1
			——[MOV	KO	D508	н	Sets the MV lower limit to "0"	
			——[Mov	K2000	D509	н	Sets the MV upper limit to "2000"	
			——[MOV	K2000	D510	Н	Sets the MV change rate limit to "2000"	
			——[MOV	K2000	D511	Ч	Sets the PV change rate limit to "2000"	

9. COMPLETE DERIVATIVE PID CONTROL INSTRUCTIONS AND PROGRAM EXAMPLES

MELSEC-Q/QnA

م ت	SM402			Гмом	F 1	DE 19	Ч	Sets the operation expression)
25				[Mov	K1	D512	Γ	for reverse operation	
				—[MOV	K 100	D513	Ъ	Sets the sampling cycle to "1 s"	
				[Mov	K100	D514	н	Sets the proportional constant to "1"	
			-[MOV	K30000		D515	Ъ	Sets the integral constant to "3000 s"	
				[Mov	KO	D516	Ъ	Sets the derivative constant to "0 s"	Setting of PID
				—[MOV	KO	D517	Ъ	Sets the filter coefficient to "0 %"	control data for loop 2
				—[MOV	KO	D518	Ъ	Sets the MV lower limit to "0"	
				[MOV	K2000	D519	Ъ	Sets the MV upper limit to "2000"	
				[MOV	K2000	D520	н	Sets the MV change rate limit to "2000"	
				[MOV	K2000	D521	н	Sets the PV change rate limit to "2000"	J
46	SM402			[PIDIN	IT	D500	Ъ	Sets the PID control data that are set in D500 to D521	
49	SM402			[Mov	KO	D600	Ъ	Sets the initial processing flag to "0"	Setting of common data of I/O data
				[Mov	K600	D610	н	Sets the SV to "600"	Setting of I/O
				—[моv	KO	D615	Ъ	Sets the automatic mode	data for loop 1
				[Mov	K1000	D628	Н	Sets the SV to "1000"	Setting of I/O
				[MOV	KO	D633	Ъ	Sets the automatic mode	data for loop 2
60	×0 				[SET	MO	Ъ	PID operation start command	
62	<u>мо</u> —- —	FROM	H8	K10	D100	K2	*Ъ	Reading the PV from the A68AD	
				[MOV	D100	D611	Ъ	Sets the PV in the I/O data area (for loop 1)	
				[MOV	D101	D629	Ъ	Sets the PV in the I/O data area (for loop 2)	
72	<u>мо</u> 			—[PIDCO	NT	D600	<u></u> -г	PID operation	

REMARK

*: It is also possible to create a program by using special function module devices. In this case the format in the ladder is as follows:



9. COMPLETE DERIVATIVE PID CONTROL INSTRUCTIONS AND PROGRAM EXAMPLES

MO Turns the A62DA output -[SET YOBB Ъ 75 enable signal ON D612 D110 Ъ -[MOV Writes the MV values of loop 1 D111 Ъ -[MOV D630 and loop 2 to the A62DA -**[T**0 HOA KO D110 K2 ŀ X1 Stops PID operation 86 -[RST MO Ъ P -{ mov X2 MO Sets the initial screen KO D200 Ъ 88 display request Monitors PID control status with the AD57 -[PID57 HOC K1 D200 Ъ

REMARK

*: It is also possible to create a program by using special function module devices. In this case the format in the ladder is as follows:

MELSEC-Q/QnA



9.3.3 Program example for changing the PID control mode between automatic and manual

An example program for switching between automatic and manual modes while executing PID operation is described below.

[PROGRAMMINGCONDITIONS]

- (1) Refer to Section 9.3.1 for details on the system configuration.
- (2) PID operation is executed for 1 loop.
- (3) The sampling cycle is 1 second.

(4) The PID control data is set in the following devices	:
Common data	D500 and D501
Loop 1 data	D502 to D511

(5) The I/O data is set in the following devices:	
Common data	D600 to D609
Loop 1 data	D610 to D627

(6) The SV and MV in manual mode are set with external digital switches as follows:
SV......X30 to X3F
MV (manual control mode).....X20 to X2F

(7) The following devices are used to start an	d stop PID control and the
automatic/manual changeover command	:
PID control start command	X0
PID control stop command	X1
Monitoring command with AD57	X2
SV setting command	X3
MV setting command in manual mode	X4

Automatic/manual mode changeover command X6

(OFF: Automatic mode, ON: Manual mode)

(8) The digital values of the A68AD and A62DA are set within the range 0 to 2000.

(9) The PID bumpless processing flag, SM774, is set to OFF. The SV is automatically rewritten to the PV when the control mode is changed from automatic to manual. Therefore, before returning the control mode from manual to automatic, the SV must be rewritten to the one used in the automatic mode. The SV is rewritten step-by-step 10 times as illustrated below:



The SV is rewritten using the operation method illustrated below:



The incremental value obtained with the formula above is added to SV every second. The remainder is added in the first addition operation.

0	SM402		-[MOV	K1	D500	Н	Sets the number of loops to be used to "1"	Setting of common
			-[MOV	K 1	D501	Ъ	Sets the number of PID operation execution loops per scan to "1"	data of PID control data
			-[MOV	KO	D502	Н	Sets the operation expression to forward operation	
			-[MOV	K100	D503	н	Sets the sampling cycle to "1 s"	
			-[MOV	K100	D504	н	Sets the proportional constant to "1"	
		[Mov	K30000		D505	Н	Sets the integral constant to "3000 s"	
			-[MOV	KO	D506	н	Sets the derivative constant to "0 s"	Setting of PID
			-[MOV	KO	D507	Ъ	Sets the filter coefficient to "0 %"	control data for loop 1
			-[MOV	KO	D508	н	Sets the MV lower limit to "0"	
			-[MOV	K2000	D509	н	Sets the MV upper limit to "2000"	
			-[MOV	K2000	D510	Ъ	Sets the MV change rate limit to "2000"	
			-[MOV	K2000	D511	н	Sets the PV change rate limit to "2000"	
25	SM402		-[PIDIN	IT	D500	н	Sets the PID control data that are set in D500 to D511	
28	SM402		-[Mov	KO	D600	н	Sets the initial processing flag to "0"	Setting of I/O
31	X3	·	-[BIN	K4X30	D610	н	Inputs the SV externally	data
			-[MOV P	D610	D200	н	Saves the SV for manual to automatic mode change processing	
38	X0 			[SET	MO	н	PID operation start command	
40	мо 	{ FROM H8	K10	D611	K1	Ъ	Sets the PV from the A68AD to	o the I/O data area.
46				MOV K	1 D	615]-	Sets the manual mode	
	-			(RST M	H4 }-		
	_			(RST M	141)-	Resets the devices used for manual to automatic mode change processing	
				[RST C	n 1-		Processing in manual mode
57	MO X6 X4	11	{	BIN K	4X20 D	050 J-	Sets the MV externally	
		└ _【 >= K2000	D50 H	(MOV ^P D)	50 D	614]		J



MEMO

_
<u></u>
 <u> </u>
 _
_

APPENDIX

Appendix 1 Processing Time List

⁽¹⁾ The following table indicates the processing times for incomplete derivative PID control instructions.

Instruction	Conditions			Processing Time (µs)							
Name	CON	ullions	Q00JCPU	Q00CPU	Q01CPU	Q02CPU	QnHCPU	QnPRHCPU			
	1	loop	115.0	97.0	88.5	64.5	28.0	28.0			
S.PIDINIT	81	oops	250.0	210.0	190.0	_	—	—			
	32	loops	—	_	_	410.0	180.0	180.0			
	1 1000	First	395.0	335.0	300.0	215.0	92.0	92.0			
	1 loop	Others	350.0	300.0	270.0	190.0	81.5	81.5			
S.PIDCONT	8 loop	First	2250.0	1850.0	1700.0	_	—	—			
S.PIDCONT		Others	1950.0	1650.0	1500.0	_	—	—			
	22 Jaana	First	—	_	_	4550.0	1950.0	1950.0			
	32 loops	Others	—	_	_	4450.0	1850.0	1850.0			
S.PIDSTOP S.PIDRUN	1 loop		79.5	66.0	61.0	25.0	11.0	11.0			
S.PIDPRMW	1 loop		120.0	99.5	89.5	60.0	26.0	26.0			

Instruction				Processing Time (µs)								
Name			Q02	JCPU	Q03UD	CPU	Q04UDHCPU,Q06UDHCPU					
	Cor	nditions			Q03UDI	ECPU	Q13UDHCPU,	Q26UDHCPU				
	COI						Q04UDEHCPU,	Q06UDEHCPU				
							Q13UDEHCPU,	Q26UDEHCPU				
			Min.	Max.	Min.	Max.	Min.	Max.				
	1	loop	14.2	49.4	14.9	22.1	11.4	18.8				
S.PIDINIT	8	8 loops					_					
	32	loops	230.1	317.3	238.2	267.9	163.9	190.9				
	1 1000	First	13	37.4	53.	5	47	.8				
	1 loop	Others	13	5.4	51.	5	45	.8				
S.PIDCONT		First	-		_		—					
5.PIDCONT	8 loop	Others	-		_		—					
	32 loops	First	11	128	104	4	931.8					
	32 100ps	Others	89	97.4	851	.5	753.8					
S.PIDSTOP	1 loop		5.6	18.5	5.5	7.6	4.9	7.0				
S.PIDRUN	1 loop		4.9	10.7	4.8	6.2	4.3	5.6				
S.PIDPRMW	1	loop	13.3	33.8	13.0	16.8	10.7	14.5				

APP

PIDCONT					Proc	essing Tir	ne (µs)				
	Cond	litions	Q2ASCPU Q2ACPU(S1)	Q3ACPU	Q2ASHCPU(S1) Q4ACPU Q4ARCPU	Q00J CPU	Q00 CPU	Q01 CPU	Q02 CPU	QnH CPU	QnPRH CPU
	1 lo	рор	61	46	23	66.0	56.0	50.5	26.0	11.2	11.2
PIDINIT	8 lo	ops	—	l	_	170.0	145.0	130.0	_		—
	32 loops		407	306	153				174.0	74.9	74.9
	1 1	First	211	159	80	325.0	275.0	245.0	86.6	37.3	37.3
	1 loop	Others	181	136	68	285.0	250.0	225.0	74.3	32.0	32.0
	8 loops	First	_	-	—	2000.0	1700.0	1500.0	—		—
PIDCONT		Others	_		—	1700.0	1450.0	1300.0			—
	<u></u>	First	5086	3824	1912		-	—	2102.5	904.9	904.9
	32 loops	Others	4894	3680	1840		_	—	2036.9	876.7	876.7
	1 1000	First	9629	7240	3620						
	1 loop	Others	606	456	228						
PID97	0 10 0 00	First	9669	7270	3635		_	_		_	
	8 loops	Others	3719	2796	1398						
PIDSTOP PIDRUN	1 lo	рор	11.2	8.4	4.2	22.0	18.5	17.0	4.5	1.9	1.9
PIDPRMW	1 lo	оор	36	26	13	53.0	45.0	41.0	14.6	6.3	6.3

(2) The following table indicates the processing times for complete derivative PID control instructions.

Instruction				Processing Time (µs)								
Name				JCPU	Q03UD	CPU	Q04UDHCPU,Q06UDHCPU					
	Con	ditions			Q03UDI	ECPU	Q13UDHCPU,	Q26UDHCPU				
							Q04UDEHCPU,	Q06UDEHCPU				
				•			Q13UDEHCPU,	Q26UDEHCPU				
			Min.	Max.	Min.	Max.	Min.	Max.				
1 loop		loop	7.3	27.8	8.2	12.5	6.3	10.9				
PIDINIT	8	loops	_	_	_	_	_	—				
	32	loops	122.4	169.3	128.5	133.2	98.7	122.6				
	1 1000	First	10	1.4	41.	6	36	36.8				
	1 loop	Others	9	1.5	41.	6	35.8					
PIDCONT	9 loon	First	-	_	_		—					
FIDCONT	8 loop	Others	-	_			—					
	22 Jaana	First	94	3.4	876	.5	655.8					
	32 loops	Others	79	9.4	771	.5	65	5.8				
PIDSTOP	1 loop		1.8	6.7	1.9	3.4	1.3	2.9				
PIDRUN	1	1 loop		6.7	1.9	3.2	1.5	2.7				
PIDPRMW	1	1 loop		17.6	7.3	10.4	5.9	8.9				

Appendix 2 Anti-Reset Windup Measure

A reset windup is a problem that an integral element keeps adding a deviation beyond a saturation limit. (It is also referred to as an integrator windup.)

When a reset windup occurs, integral operation must be stopped to enable immediate response to the inversion of the deviation.

Since the anti-reset windup measure is taken in the PID operation instruction (PIDCONT instruction and S.PIDCONT instruction) of the QCPU/QnACPU, it is unnecessary to stop the integral operation.



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 onsite 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.

6. Product application

- (1) In using the Mitsubishi MELSEC programmable logic controller, the usage conditions shall be that the application will not lead to a major accident even if any problem or fault should occur in the programmable logic controller device, and that backup and fail-safe functions are systematically provided outside of the device for any problem or fault.
- (2) The Mitsubishi programmable logic controller has been designed and manufactured for applications in general industries, etc. Thus, applications in which the public could be affected such as in nuclear power plants and other power plants operated by respective power companies, and applications in which a special quality assurance system is required, such as for Railway companies or Public service purposes shall be excluded from the programmable logic controller applications.

In addition, applications in which human life or property that could be greatly affected, such as in aircraft, medical applications, incineration and fuel devices, manned transportation, equipment for recreation and amusement, and safety devices, shall also be excluded from the programmable logic controller range of applications. However, in certain cases, some applications may be possible, providing the user consults their local Mitsubishi representative outlining the special requirements of the project, and providing that all parties concerned agree to the special circumstances, solely at the users discretion.

QCPU(Q Mode)/QnACPU

Programming Manual (PID Control Instructions)

QCPU(Q)-P(PI)-E

MODEL

MODEL CODE

13JF59

SH(NA)-080040-K(0805)MEE

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN NAGOYA WORKS : 1-14 , YADA-MINAMI 5-CHOME , HIGASHI-KU, NAGOYA , JAPAN

> When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.