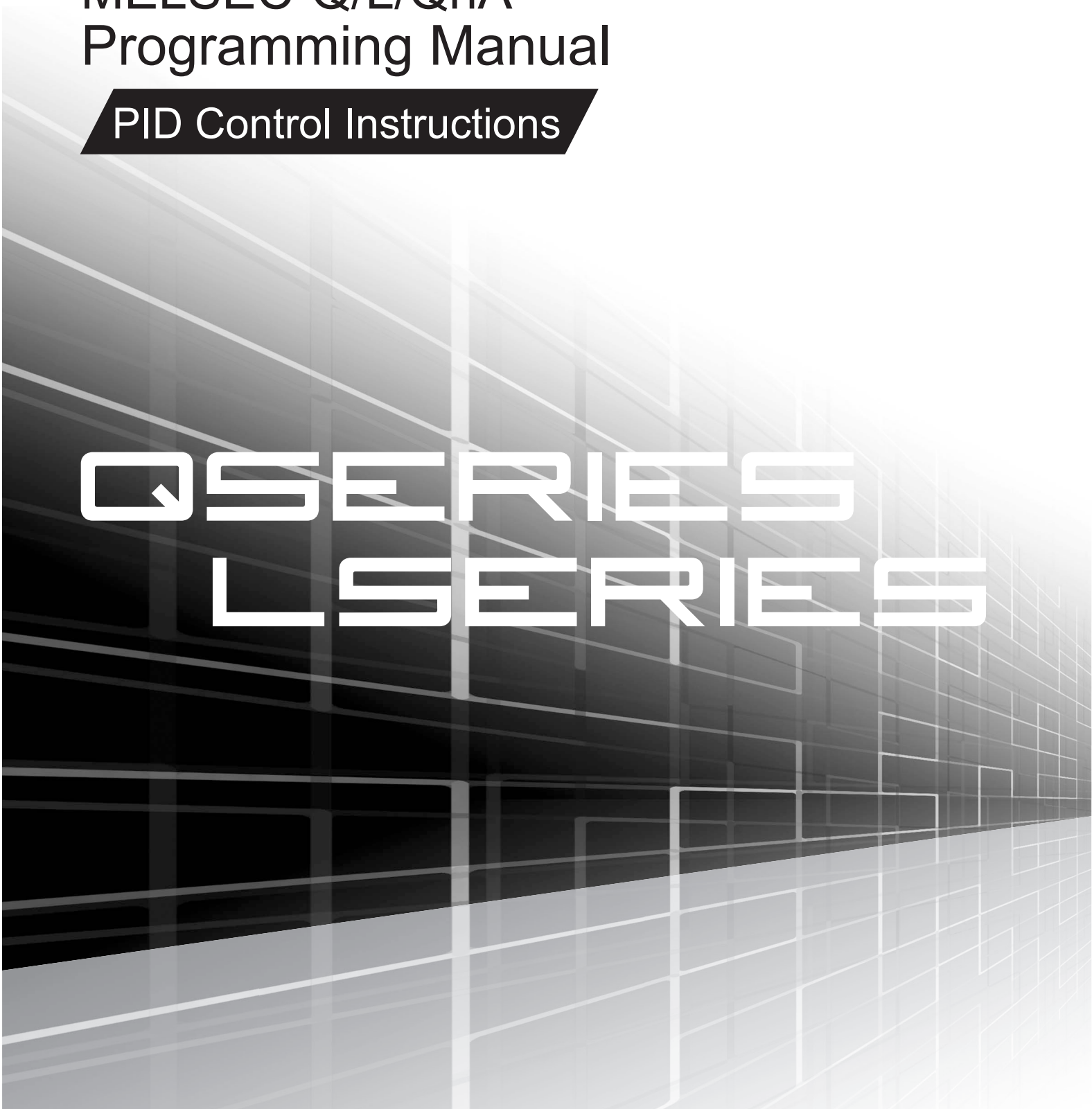
MITSUBISHI

Mitsubishi Programmable Controller

MELSEC Q series MELSEC L series

MELSEC-Q/L/QnA Programming Manual

PID Control Instructions



• 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".

⚠ WARNING	Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.
⚠ CAUTION	Indicates that incorrect handling may cause hazardous conditions, resulting in minor or moderate injury or property damage.

Note that the ⚠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]

⚠ WARNING

- Configure safety circuits external to the programming controller to ensure that the entire system operates safely even when a fault occurs in the external power supply or the programmable controller. Failure to do so may result in an accident due to an incorrect output or malfunction.
  (1) Emergency stop circuits, protection circuits, and protective interlock circuits for conflicting operations (such as forward/reverse rotations or upper/lower limit positioning) must be configured external to the programmable controller.
  (2) When the programmable controller detects an abnormal condition, it stops the operation and all outputs are:
    - Turned off if the overcurrent or overvoltage protection of the power supply module is activated.
    - Held or turned off according to the parameter setting if the self-diagnostic function of the CPU module detects an error such as a watchdog timer error.

  Also, all outputs may be turned on if an error occurs in a part, such as an I/O control part, where the CPU module cannot detect any error. To ensure safety operation in such a case, provide a safety mechanism or a fail-safe circuit external to the programmable controller. For a fail-safe circuit example, refer to "General Safety Requirements" in the MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection).
- 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]

⚠ WARNING

- When changing data from a peripheral device connected to the CPU module during operation, configure an interlock circuit in the program to ensure that the entire system will 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]

⚠ CAUTION

- 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.

• CONDITIONS OF USE FOR THE PRODUCT •

(1) Mitsubishi programmable controller ("the PRODUCT") shall be used in conditions;
i) where any problem, fault or failure occurring in the PRODUCT, if any, shall not lead to any major or serious accident; and
ii) where the backup and fail-safe function are systematically or automatically provided outside of the PRODUCT for the case of any problem, fault or failure occurring in the PRODUCT.

(2) The PRODUCT has been designed and manufactured for the purpose of being used in general industries.
MITSUBISHI SHALL HAVE NO RESPONSIBILITY OR LIABILITY (INCLUDING, BUT NOT LIMITED TO ANY AND ALL RESPONSIBILITY OR LIABILITY BASED ON CONTRACT, WARRANTY, TORT, PRODUCT LIABILITY) FOR ANY INJURY OR DEATH TO PERSONS OR LOSS OR DAMAGE TO PROPERTY CAUSED BY the PRODUCT THAT ARE OPERATED OR USED IN APPLICATION NOT INTENDED OR EXCLUDED BY INSTRUCTIONS, PRECAUTIONS, OR WARNING CONTAINED IN MITSUBISHI'S USER, INSTRUCTION AND/OR SAFETY MANUALS, TECHNICAL BULLETINS AND GUIDELINES FOR the PRODUCT.
("Prohibited Application")
Prohibited Applications include, but not limited to, the use of the PRODUCT in;
- Nuclear Power Plants and any other power plants operated by Power companies, and/or any other cases in which the public could be affected if any problem or fault occurs in the PRODUCT.
- Railway companies or Public service purposes, and/or any other cases in which establishment of a special quality assurance system is required by the Purchaser or End User.
- Aircraft or Aerospace, Medical applications, Train equipment, transport equipment such as Elevator and Escalator, Incineration and Fuel devices, Vehicles, Manned transportation, Equipment for Recreation and Amusement, and Safety devices, handling of Nuclear or Hazardous Materials or Chemicals, Mining and Drilling, and/or other applications where there is a significant risk of injury to the public or property.

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

REVISIONS

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

Print Date	* Manual Number	Revision
Dec., 1999	SH (NA) 080040-A	First edition
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
Dec., 2008	SH (NA) 080040-L	Addition of Universal model QCPU Addition module Q00UJCPU, Q00UCPU, Q01UCPU, Q10UDHCPU, Q10UDEHCPU, Q20UDHCPU, Q20UDEHCPU Partial correction Related Manuals, Generic Terms and Abbreviations Used in This Manual, Section 2.1, Section 3.1.3, Appendix 1
Jan., 2010	SH (NA) 080040-M	Revision owing to addition of the MELSEC-L series Partial correction SAFETY PRECAUTIONS, About Manuals, GENERIC TERMS, Chapter 1, Chapter 2, Section 2.1, Section 3.1.1, Section 3.1.3, Section 3.2.1, Section 3.2.3, Section 4.3.5, Section 5.1, Section 5.2, Chapter 6, Chapter 7, Section 8.1.1, Section 8.1.2, Section 8.1.3, Section 8.1.4, Section 9.1.1, Section 9.1.2, Section 9.1.3, Section 9.1.4, Section 9.1.5, Section 9.2, Appendix 1, Appendix 2, Addition module CONDITIONS OF USE FOR THE PRODUCT

Japanese Manual Version SH-080022-M

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/L/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
CONDITIONS OF USE FOR THE PRODUCT ..... A- 3
REVISIONS ..... A- 4
INTRODUCTION ..... A- 6
CONTENTS ..... A- 6
About Manuals ..... A- 9
GENERIC TERMS A-12

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
3.1.2 PID operation block diagram and operation expressions ..... 3 - 2
3.1.3 PID control instruction list ..... 3 - 3
3.2 PID Control by Complete Derivative ..... 3 - 8
3.2.1 Performance specifications ..... 3 - 8
3.2.2 PID operation block diagram and operation expressions ..... 3 - 9
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
4.2.1 Operation method ..... 4 - 2
4.2.2 Forward operation and reverse operation ..... 4 - 2
4.2.3 Proportionate operation (P operation) ..... 4 - 4
4.2.4 Integrating operation (I operation) ..... 4 - 5
4.2.5 Differentiating operation (D operation) ..... 4 - 6
4.2.6 PID operation ..... 4 - 8
4.3 Other Functions ..... 4 - 9
4.3.1 Bumpless changeover function ..... 4 - 9
4.3.2 MV higher/lower limit control function ..... 4 - 10
4.3.3 Monitorning PID control with the AD57(S1) (QnACPU only) ..... 4 - 11
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 (for QCPU and LCPU) ..... 4 - 13

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 - 16
5.2 I/O Data ..... 5 - 18

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 ..... 8 - 1
8.1.1 PID control data settings S.PIDINIT,SP.PIDINIT ..... 8 - 2
8.1.2 PID operation S.PIDCONT,SP.PIDCONT ..... 8 - 3
8.1.3 Operation stop/start of designated loop no. S.PIDSTOP,SP.PIDSTOP,S.PIDRUN,SP.PIDRUN ..... 8 - 5
8.1.4 Parameter change at designated loop S.PIDPRMW,SP.PIDPRMW ..... 8 - 6
8.2 PID Control Program Examples ..... 8 - 8
8.2.1 System configuration for program examples ..... 8 - 8
8.2.2 Program example for automatic mode PID control ..... 8 - 9
8.2.3 Program example for changing the PID control mode between automatic and manual ..... 8 - 13

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 (for QCPU and LCPU) ..... 9 - 11
9.2.1 System configuration for program examples ..... 9 - 11
9.2.2 Program example for automatic mode PID control ..... 9 - 12
9.2.3 Program example for changing the PID control 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 control ..... 9 - 20
9.3.3 Program example for changing the PID control 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 .....APP - 3

About Manuals

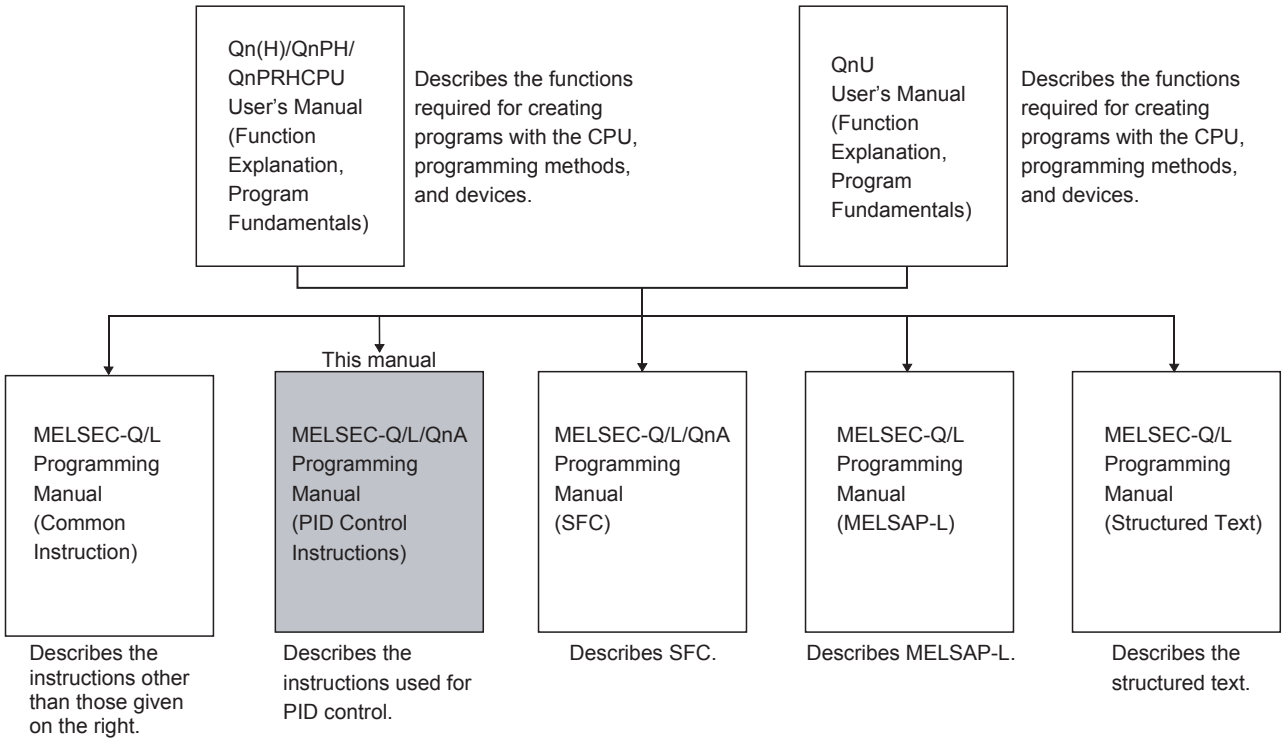
The following manuals are also related to this product.
In necessary, order them by quoting the details in the tables below.

Related Manuals

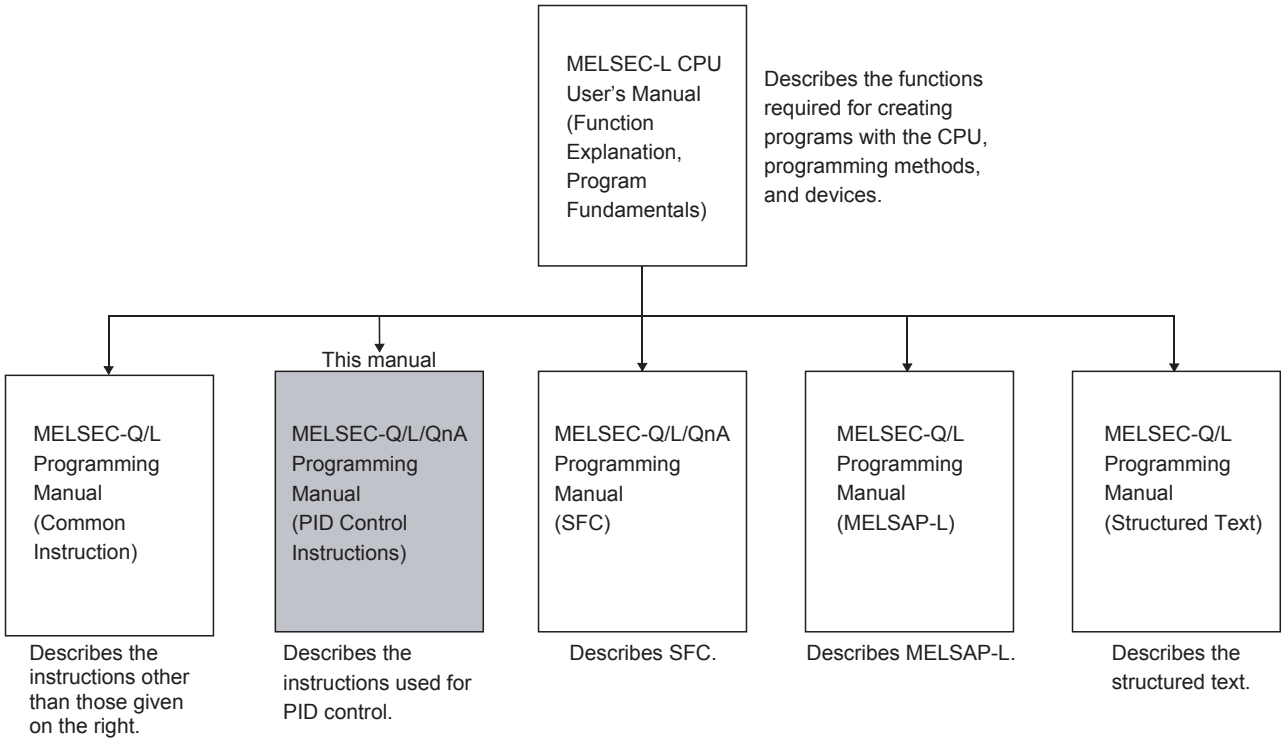
Manual Name	Manual Number (Model Code)
QnUCPU User's Manual (Function Explanation, Program Fundamentals) Describes the functions, programming procedures, devices, etc. necessary to create programs. (Sold separately)	SH-080807ENG (13JZ27)
Qn(H)/QnPH/QnPRHCPU Users Manual(Function Explanation, Program Fundamentals) Describes the functions, programming procedures, devices, etc. necessary to create programs. (Sold separately)	SH-080808ENG (13JZ28)
MELSEC-L CPU Module User's Manual (Function Explanation, Program Fundamentals) Describes the functions, programming procedures, devices, etc. necessary to create programs. (Sold separately)	SH-080889ENG (13JZ35)
QnACPU Programming Manual (Fundamentals) Describes how to create programs, the names of devices, parameters, and types of program. (Sold separately)	IB-66614 (13JF46)
MELSEC-Q/L Programming Manual (Common Instructions) Describes how to use sequence instructions, basic instructions, and application instructions. (Sold separately)	SH-080809ENG (13JW10)
QnACPU Programming Manual (Common Instructions) Describes how to use sequence instructions, basic instructions, and application instructions. (Sold separately)	SH-080810ENG (13JW11)
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)
MELSEC-Q/L/QnA Programming Manual (SFC) Describes the system configuration, performance specifications, functions, programming, debugging, and error codes for SFC (MELSAP3) programs. (Sold separately)	SH-080041 (13JF60)
MELSEC-Q/L Programming Manual (MELSAP-L) Describes programming methods, specifications, and functions for SFC (MELSAP-L) programs. (Sold separately)	SH-080076 (13JF61)
MELSEC-Q/L Programming Manual (Structured Text) Describes programming using the structured text language. (Sold separately)	SH-080366E (13JF68)

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.

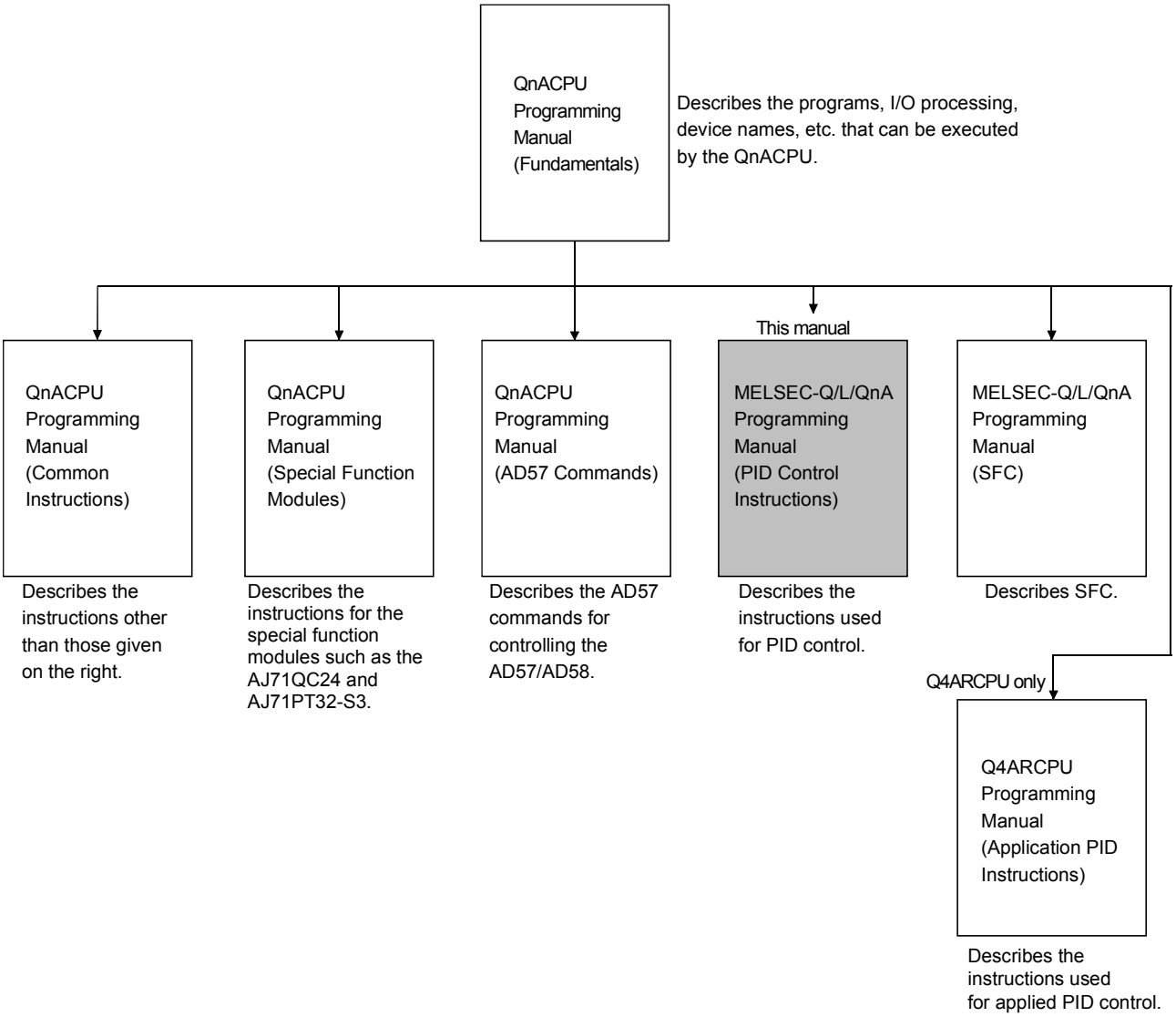
(1) When QCPU is used



(2) When LCPU is used



(3) When QnACPU is used



GENERIC TERMS

This manual uses the following generic terms unless otherwise described.

Generic term	Description
CPU module	Generic term for Basic model QCPU, High Performance model QCPU, Redundant CPU, Universal model QCPU, LCPU, and QnACPU
QCPU	Generic term for Q00CPU, Q01CPU, Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU, Q12PRHCPU, Q25PRHCPU, Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU
QnCPU	Generic term for Q02CPU
QnHCPU	Generic term for Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU
QnPHCPU	Generic term for Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU
QnPRHCPU	Generic term of Q12PRHCPU, Q25PRHCPU
LCPU	Generic term for L02CPU and L26CPU-BT
QnACPU	Generic term for Q2ASCPU, Q2ASCPU-S1, Q2ASHCPU, Q2ASHCPU-S1, Q2ACPU, Q2ACPU-S1, Q3ACPU, Q4ACPU, Q4ARCPU
QnA	Generic term for Q2ASCPU, Q2ASCPU-S1, Q2ASHCPU, Q2ASHCPU-S1, Q2ACPU, Q2ACPU-S1, Q3ACPU, Q4ACPU
Q4AR	Generic term of Q4ARCPU
QnUD(H)CPU	Generic term for Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q13UDEHCPU, Q26UDEHCPU
Basic model QCPU	Generic term for Q00JCPU, Q00CPU, Q01CPU
Basic	
High Performance model QCPU	Generic term for Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU
High Performance	
Process CPU	Generic term for Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU
Redundant CPU	Generic term for Q12PRHCPU, Q25PRHCPU
Universal model QCPU	Generic term for Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU
Universal	

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
- LCPU
- QnACPU

For the Basic model QCPU, High Performance model QCPU, Redundant CPU, Universal model QCPU, and LCPU, the instructions for PID control by using complete and incomplete derivatives are featured as standard.

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 Module Model Name		Incomplete Derivative	Complete Derivative
Basic model QCPU	First five digits of serial No. are "04121" or earlier	×	×
	First five digits of serial No. are "04122" or later	○	○*1
High Performance model QCPU	First five digits of serial No. are "05031" or earlier	×	○
	First five digits of serial No. are "05032" or later	○	○
Redundant CPU		○	○
Universal model QCPU		○	○
LCPU		○	○
QnACPU		×	○

○: Usable, ×: 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.

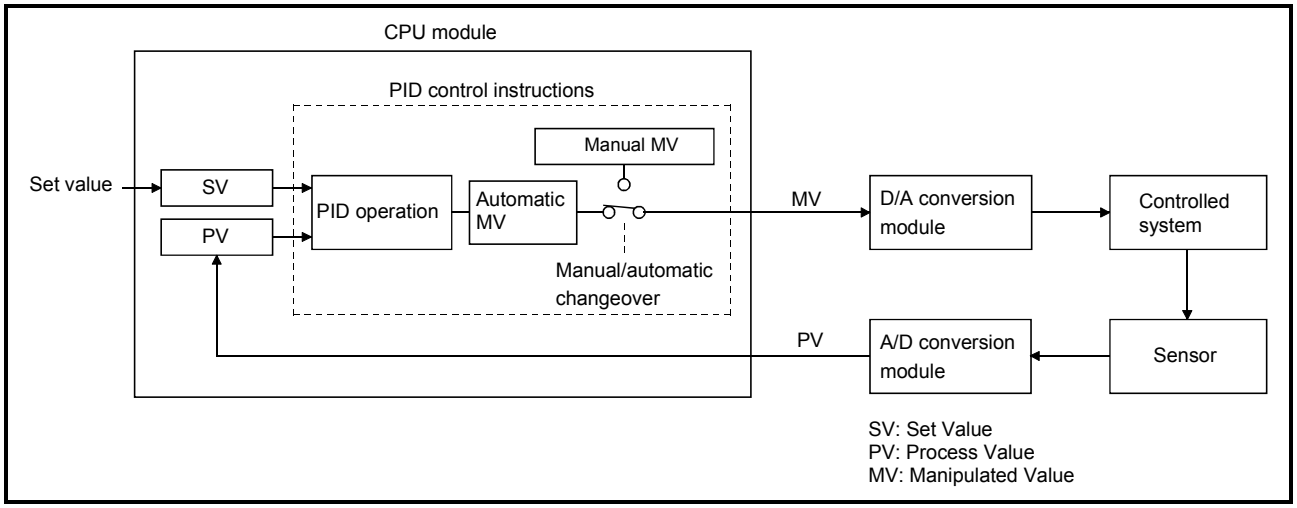


Figure 1.1 Overview of PID Control Processing

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.

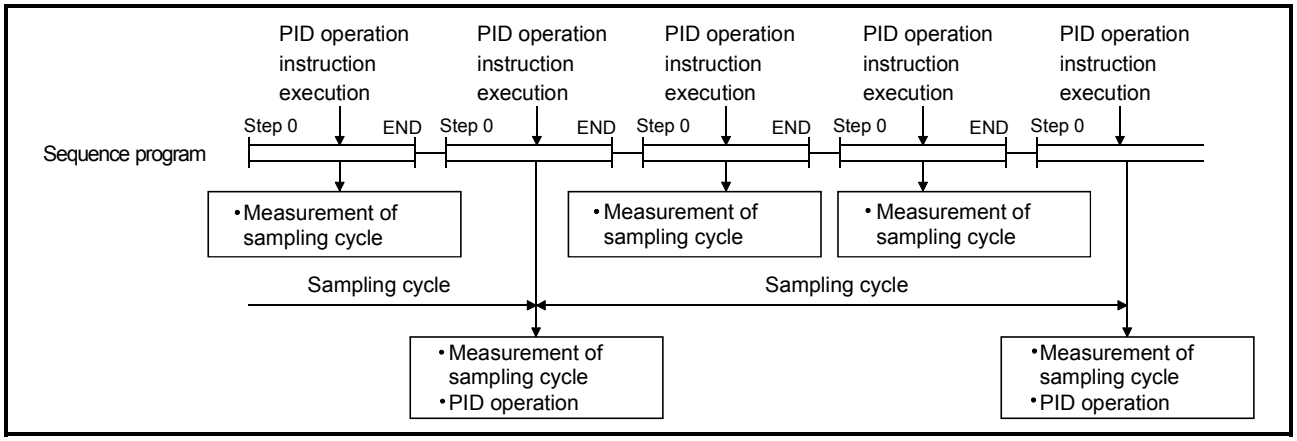


Figure 1. 2 Operation when PID Operation Instruction Executed

REMARK

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

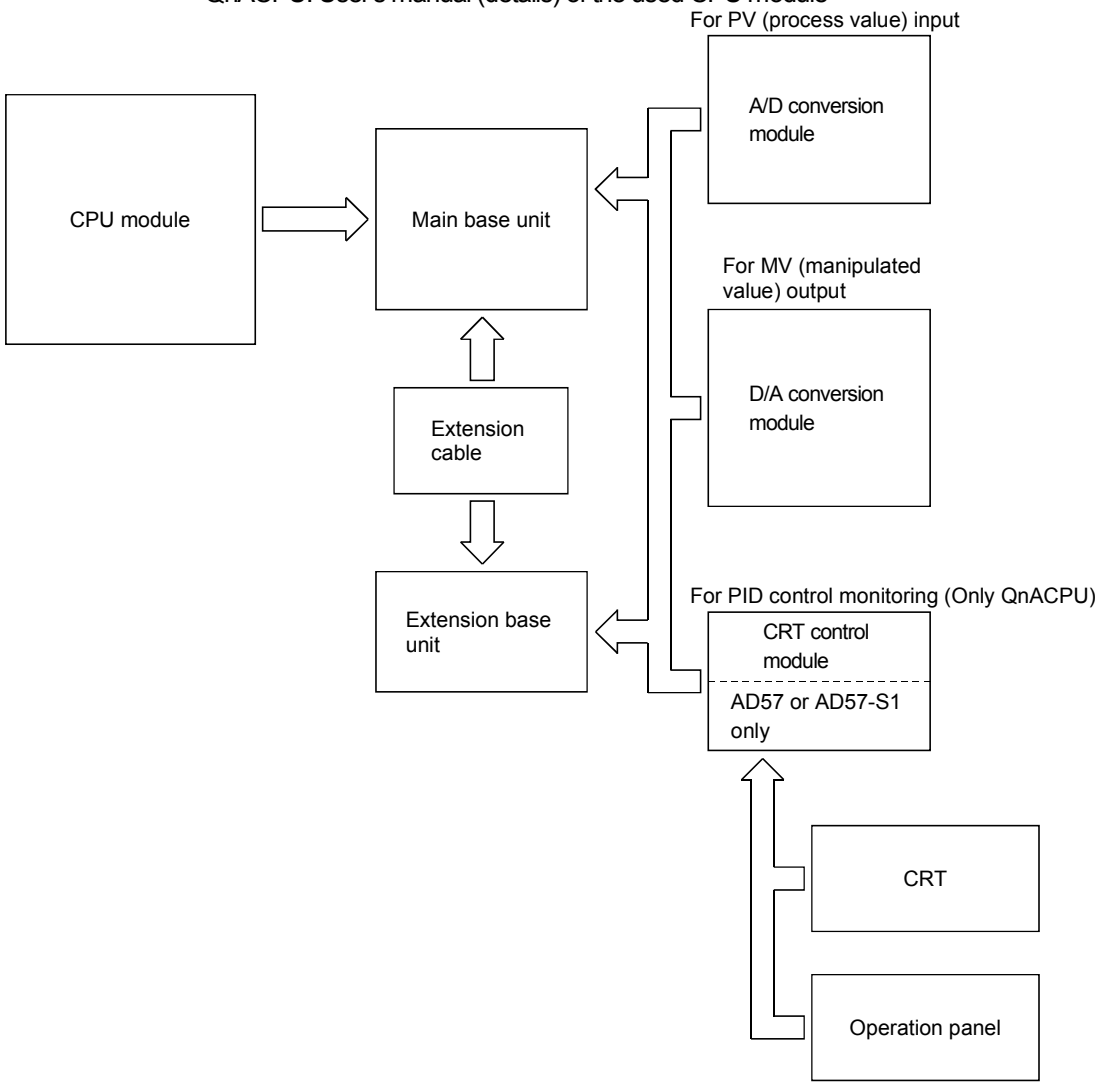
MEMO

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	
	0 to 2000 fixed *	Any setting
Basic model QCPU	○	○
High Performance model QCPU	○	○
Redundant CPU	○	○
Universal model QCPU	○	○
LCPU	○	○
QnACPU	○	×

○: Can be set, ×: Cannot be set

*: When the resolution of the A/D converter module or D/A converter module used for I/O of PID control is other than 0 to 2000, convert the digital 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
Universal model QCPU	Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU
LCPU	L02CPU, L26CPU-BT
QnACPU	Q2ASCPU, Q2ASCPU-S1, Q2ASHCPU, Q2ASHCPU-S1 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

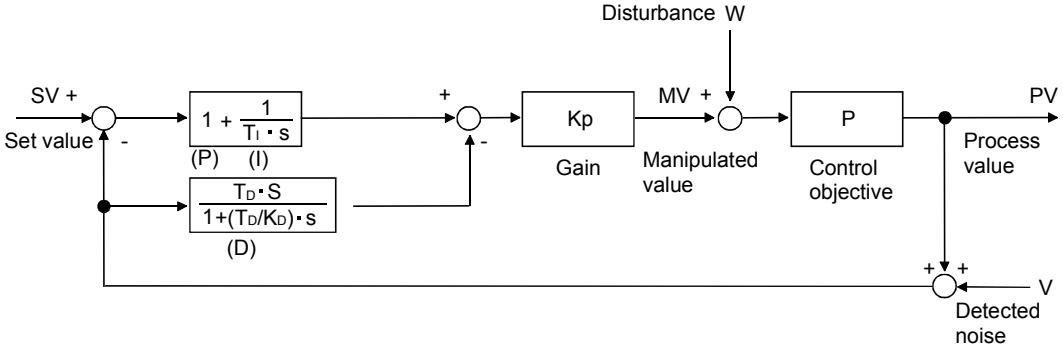
The performance specifications for PID control are tabled below.

Item			Specifications				
			With PID limits		Without PID limits		
			Basic model QCPU	High Performance model QCPU, Redundant CPU, Universal model QCPU LCPU	Basic model QCPU	High Performance model QCPU, Redundant CPU, Universal model QCPU LCPU	QnA CPU
Number of PID control loops		—	8 loops (maximum)	32 loops (maximum)	8 loops (maximum)	32 loops (maximum)	—
Sampling cycle		T_S	0.01 to 60.00 s				—
PID operation method		—	Process value differentiation incomplete derivative (forward operation/reverse operation)				—
PID constant setting range	Proportional constant	K_P	0.01 to 100.00				—
	Integral constant	T_I	0.1 to 3000.0 s				—
	Derivative constant	T_D	0.00 to 300.00 s				—
	Derivative gain	K_D	0.00 to 300.00				—
SV (set value) setting range		SV	0 to 2000		-32768 to 32767		—
PV (process value) setting range		PV	-50 to 2050		-32768 to 32767		—
MV (manipulated value) output range		MV					—

—: Unusable

3.1.2 PID operation block diagram and operation expressions

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



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

Name		Operation Expressions	Meanings of Symbols
Process value differentiation Incomplete derivative	Forward operation	$EV_n=PV_{fn}^*-SV$ $\Delta MV=K_P\{(EV_n-EV_{n-1})+\frac{T_S}{T_I}EV_n+D_n\}$ $D_n=\frac{T_D}{T_S+\frac{T_D}{K_D}}(PV_{fn}-2PV_{fn-1}+PV_{fn-2})+\frac{\frac{T_D}{K_D}}{T_S+\frac{T_D}{K_D}}D_{n-1}$ $MV_n=\Sigma\Delta MV$	EV_n : Deviation in the present sampling cycle EV_{n-1} : Deviation in the preceding sampling cycle SV : Set value PV_{fn} : Process value of the present sampling cycle (after filtering) PV_{fn-1} : Process value of the preceding sampling cycle (after filtering) PV_{fn-2} : Process value of the sampling cycle two cycles before (after filtering) $\triangle MV$: Output change value MV_n : Present manipulation value D_n : Present derivative term D_{n-1} : Derivative term of the preceding sampling cycle K_P : Proportional constant T_S : Sampling cycle T_I : Integral constant T_D : Derivative constant K_D : Derivative gain
	Reverse operation	$EV_n=SV-PV_{fn}^*$ $\Delta MV=K_P\{(EV_n-EV_{n-1})+\frac{T_S}{T_I}EV_n+D_n\}$ $D_n=\frac{T_D}{T_S+\frac{T_D}{K_D}}(-PV_{fn}+2PV_{fn-1}-PV_{fn-2})+\frac{\frac{T_D}{K_D}}{T_S+\frac{T_D}{K_D}}D_{n-1}$ $MV_n=\Sigma\Delta MV$	

POINT

(1) *:PV_{fn} is calculated using the following expression.
Therefore, it 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 $PV_{fn}= PV_n+\alpha (PV_{fn-1}-PV_n)$

PV_n : 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)

3.1.3 PID control instruction list

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

Instruction Name	Processing Details	CPU	
		QCPU, LCPU	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.	○	×
S.PIDPRMW	Changes the operation parameters for the designated loop number to PID control data.	○*	×

○: Usable, ×: Unusable

*: For the Basic model QCPU, High Performance model QCPU, Redundant CPU, Universal model QCPU, and LCPU, "with or without PID limits" can be selected. 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

Category	Instruction Symbol	Ladder Format	Processing Details	Excution Condition	Number of Basic Steps	Subset Processing	Page
Contril data setting	S.PIDINIT	S.PIDINIT Ⓢ SP.PIDINIT Ⓢ	Sets the PID control data stored in the word device (designated by Ⓢ) Ⓢ+0 Common data setting area Ⓢ+1 Ⓢ+2 to Ⓢ+15 For loop 1 Ⓢ+16 to Ⓢ+29 For loop 2 to Ⓢ+(m+0) to Ⓢ+(m+13) For loop n m=(n-1)×14+2		7	—	8-2

(1) (2) (3) (4) (5) (6) (7) (8)

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.

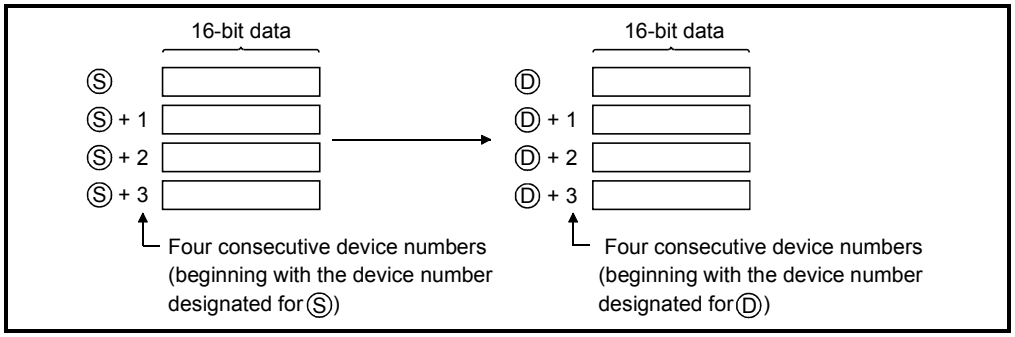


Fig. 3.1 Processing for Each Instruction

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

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.

(6) Number of instruction steps

For details on the number of steps, refer to the Programming Manual (Common Instructions) for the CPU module used.

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

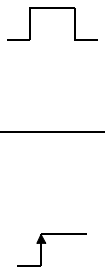
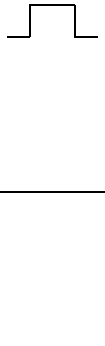
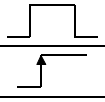
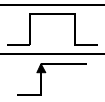
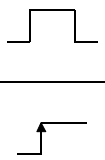
－ indicates that subset processing is impossible.

For details on subset processing, refer to the Programming Manual (Common Instructions) for the CPU module used.

(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 data setting	S.PIDINIT	S.PIDINIT Ⓢ SP.PIDINIT Ⓢ	Sets the PID control data stored in the word device (designated by Ⓢ). Ⓢ + 0 Common data setting area Ⓢ + 1 Ⓢ + 2 to Ⓢ + 15 For loop 1 Ⓢ + 16 to Ⓢ + 29 For loop 2 to Ⓢ + (m+0) to Ⓢ + (m+13) For loop n m=(n-1) × 14+2		7	—	8-2
PID operation	S.PIDCONT	S.PIDCONT Ⓢ SP.PIDCONT Ⓢ	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 Ⓢ. Ⓢ to Ⓢ + 9 Common data setting area Ⓢ + 10 SV setting area PV setting area to MV value storage area For loop 1 Ⓢ + 32 Ⓢ + 33 SV setting area PV setting area to MV value storage area For loop 2 Ⓢ + 55 Ⓢ + (m+0) SV setting area PV setting area to MV value storage area For loop n Ⓢ + (m+22) m=(n-1) × 23+10		7	—	8-3
Operation stop	S.PIDSTOP	S.PIDSTOP Ⓝ SP.PIDSTOP Ⓝ	Stops the PID operation at the loop number designated by Ⓝ.		7	—	8-5
Operation start	S.PIDRUN	S.PIDRUN Ⓝ SP.PIDRUN Ⓝ	Starts the operation at the loop number designated by Ⓝ.		6	—	8-5
Parameter change	S.PIDPRMW	S.PIDPRMW Ⓝ Ⓢ SP.PIDPRMW Ⓝ Ⓢ	Changes the operation parameter for the loop number designated by Ⓝ to the PID control data stored in the word device designated by Ⓢ		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

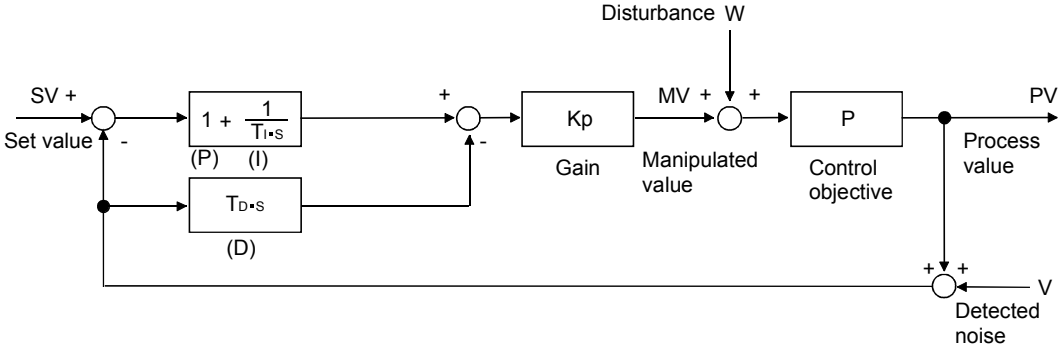
3.2.1 Performance specifications

The performance specifications for PID control are tabled below.

Item			Specification				
			With PID limits		Without PID limits		QnACPU
			Basic model QCPU	High Performance model QCPU, Redundant CPU, Universal model QCPU, LCPU	Basic model QCPU	High Performance model QCPU, Redundant CPU, Universal model QCPU, LCPU	
Number of PID control loops		—	8 loops (maximum)	32 loops (maximum)	8 loops (maximum)	32 loops (maximum)	32 loops (maximum)
Sampling cycle		T_S	0.01 to 60.00 s				
PID operation method		—	Process value differentiation complete derivative (forward operation/reverse operation)				
PID constant setting range	Proportional constant	K_P	0.01 to 100.00				
	Integral constant	T_I	0.1 to 3000.0 s				
	Derivative constant	T_D	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		-32768 to 32767		-50 to 2050
MV (manipulated value) output range		MV					

3.2.2 PID operation block diagram and operation expressions

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



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

Name		Operation Expressions	Meanings of Symbols
Process value differentiation Complete derivative	Forward operation	$EV_n=PV_{fn}^*-SV$ $\Delta MV=K_P\{(EV_n-EV_{n-1})+\frac{T_S}{T_I}EV_n+D_n\}$ $D_n=\frac{T_D}{T_S}\cdot(PV_{fn}-2PV_{fn-1}+PV_{fn-2})$ $MV_n=\Sigma\Delta MV$	EV_n : Deviation in the present sampling cycle EV_{n-1} : Deviation in the preceding sampling cycle SV : Set value PV_{fn} : Process value of the present sampling cycle (after filtering) PV_{fn-1} : Process value of the preceding sampling cycle (after filtering) PV_{fn-2} : Process value of the sampling cycle two cycles before (after filtering) $\triangle MV$: Output change value MV_n : Present manipulation value D_n : Present derivative term K_P : Proportional constant T_S : Sampling cycle T_I : Integral constant T_D : Derivative constant
	Reverse operation	$EV_n=SV-PV_{fn}^*$ $\Delta MV=K_P\{(EV_n-EV_{n-1})+\frac{T_S}{T_I}EV_n+D_n\}$ $D_n=\frac{T_D}{T_S}(-PV_{fn}+2PV_{fn-1}-PV_{fn-2})$ $MV_n=\Sigma\Delta MV$	

POINT

(1) *:PV_{fn} is calculated using the following expression.
Therefore, it 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 $PV_{fn}= PV_n+\alpha (PV_{fn-1}-PV_n)$

PV_n : 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)

3.2.3 PID control instruction list

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

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

○: Usable, ×: Unusable

*: For the Basic model QCPU, High Performance model QCPU, Redundant CPU, Universal model QCPU, and LCPU, "with or without PID limits" can be selected. 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

Category	Instruction Symbol	Ladder Format	Processing Details	Excution Condition	Number of Basic Steps	Subset Processing	Page
PID control data setting	PIDINIT	PIDINIT Ⓢ	Sets the PID control data stored in the word device (designated by Ⓢ) Ⓢ + 0 Common data setting area Ⓢ + 1 Ⓢ + 2 to Ⓢ + 11 For loop 1 Ⓢ + 12 to Ⓢ + 21 For loop 2 to Ⓢ + (m+0) to Ⓢ + (m+9) For loop n m=(n-1) × 10+2		2	—	9-2
		PIDINITP Ⓢ					

(1) (2) (3) (4) (5) (6) (7) (8)

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.

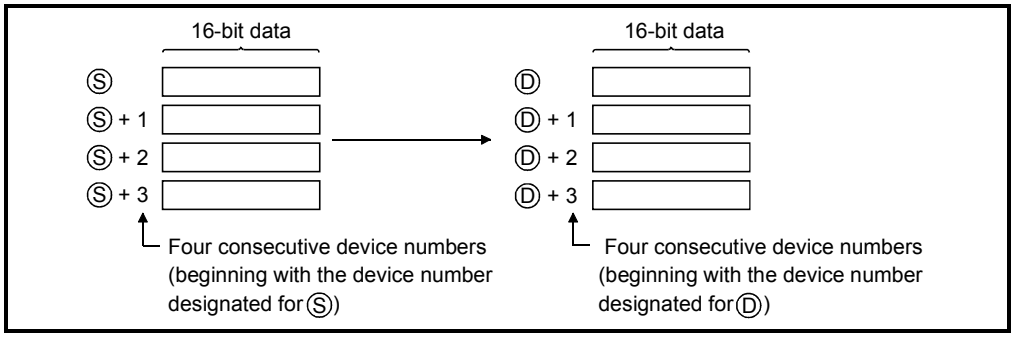


Fig. 3.2 Processing for Each Instruction

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

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.

(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).

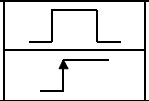
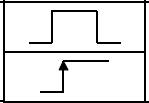
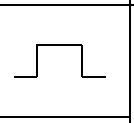
(8) Indicates the page number in this manual where a detailed description for the instruction can be found.

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 Ⓢ PIDINITP Ⓢ	Sets the PID control data stored in the word device (designated by Ⓢ). Ⓢ+ 0 Common data setting area Ⓢ+ 1 Ⓢ+ 2 to Ⓢ+ 11 For loop 1 Ⓢ+ 12 to Ⓢ+ 21 For loop 2 to Ⓢ + (m+0) to Ⓢ + (m+9) For loop n m=(n-1) × 10+2		2	—	9-2
PID operation	PIDCONT	PIDCONT Ⓢ PIDCONTP Ⓢ	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 to Ⓢ+ 9 Common data setting area Ⓢ+ 10 to Ⓢ+ 27 SV setting area / PV setting area / MV value starage area For loop 1 Ⓢ+ 28 to Ⓢ+ 45 SV setting area / PV setting area / MV value starage area For loop 2 Ⓢ+ (m+0) to Ⓢ+ (m+17) SV setting area / PV setting area / MV value starage area For loop n m=(n-1) × 18+10		2	—	9-3
Monitoring	PID57	PID57 ⓝ S1 S2 PID57P ⓝ S1 S2	Monitors the PID operation results for the AD57 (S1) (designated by ⓝ). ⓝ : First I/O number of the AD57(S1) S1 : Monitor screen number 1:Loop 1 to loop 8 2:Loop 9 to loop16 3:Loop17 to loop24 4:Loop25 to loop32 S2 : Initial screen display request		4	—	9-5

Table 3.4 PID Control Instruction List

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

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:

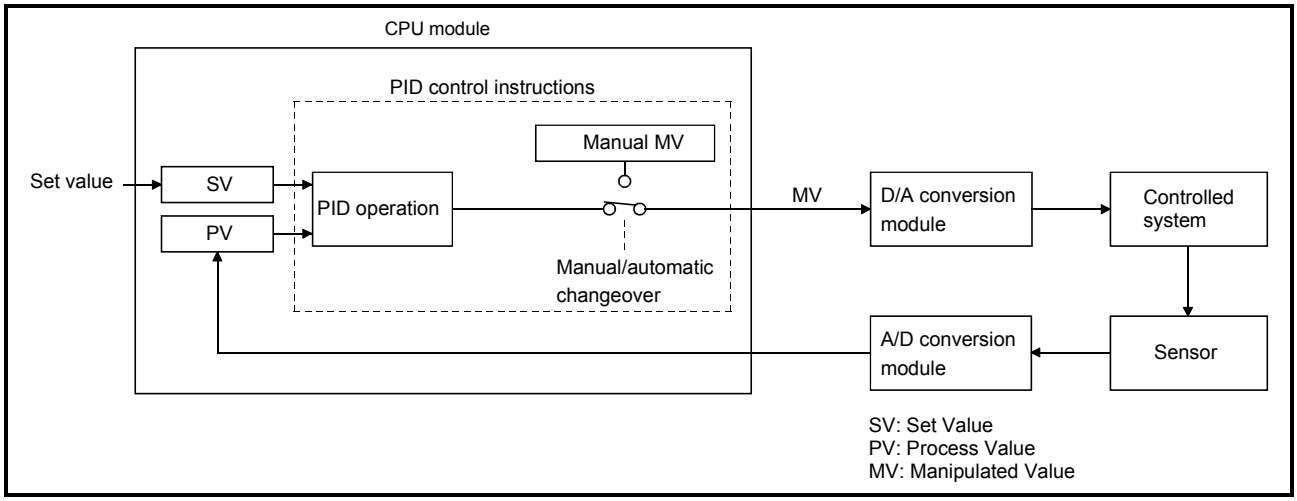


Fig. 4.1 Application of PID Control Process Control

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

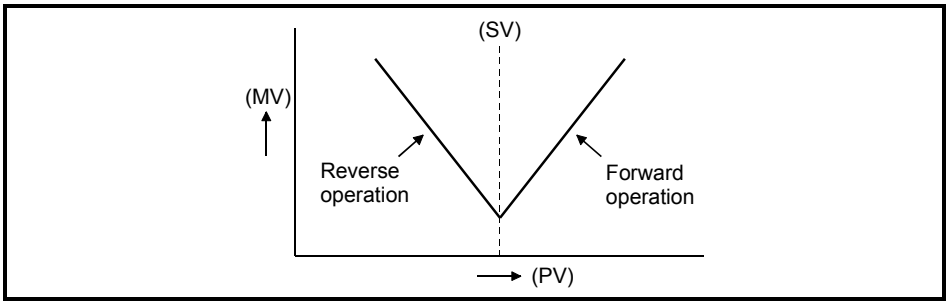
Either forward operation or reverse operation can be selected to designate the direction of PID control.

(1) In forward operation, the MV (manipulated value) increases as the PV (process value) increases beyond the SV (set value).

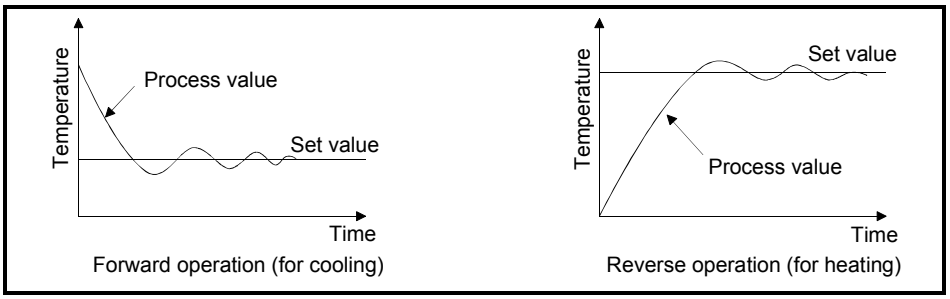
(2) In reverse operation, the MV increases as the PV decreases below the SV.

(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:

$$MV=Kp \cdot E$$

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 larger	Control operation gets faster. 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.

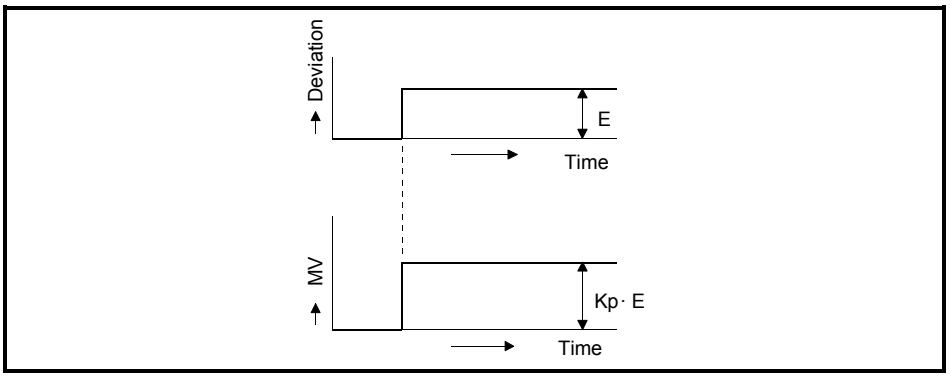
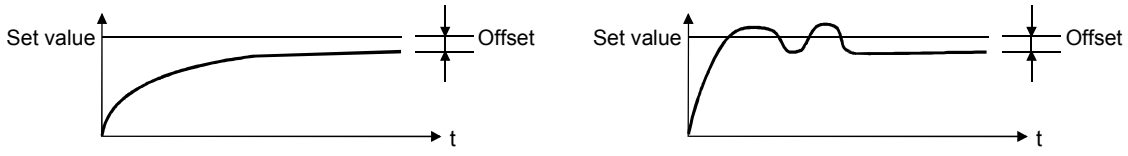


Fig. 4.2 Proportional Operation with a Constant Deviation

(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 T_I.

Condition	Integral Operation
When integral time T_I is shorter	Integrating effect increases and the time to eliminate the offset becomes shorter. However, hunting is more likely to occur.
When integral time T_I 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.

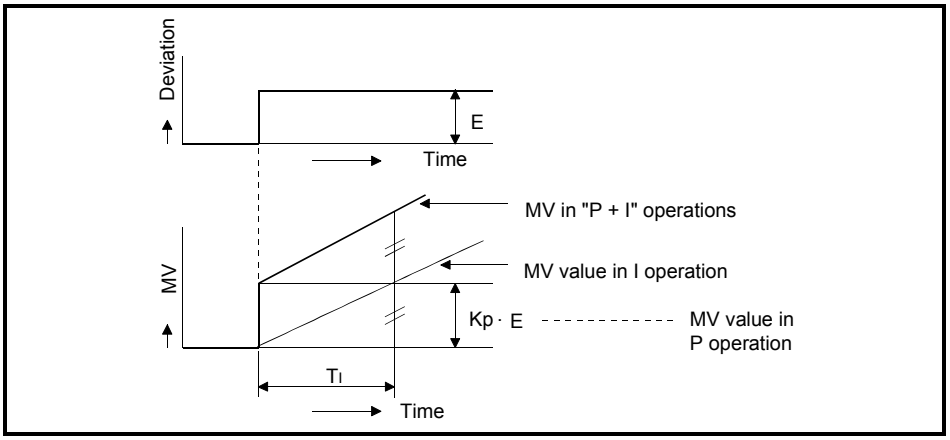


Fig. 4.3 Integral Operation with a Constant Deviation

(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.

(1) 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 T_D 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.

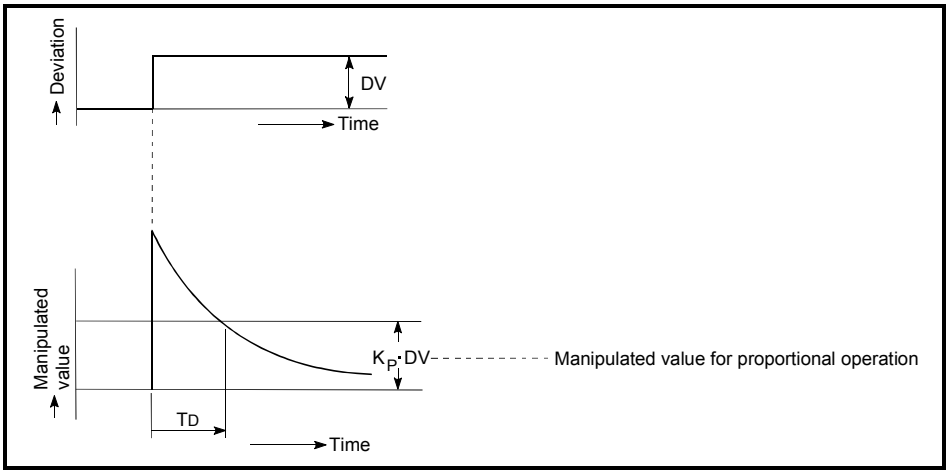


Fig. 4.4 Derivative operation when the deviation is a constant

(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.

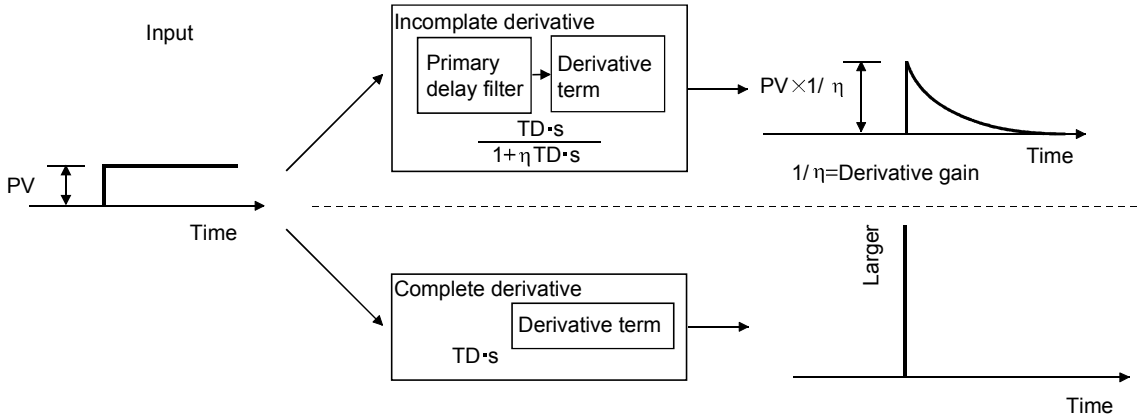
Incomplete derivative is effective for the following cases.

- 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.

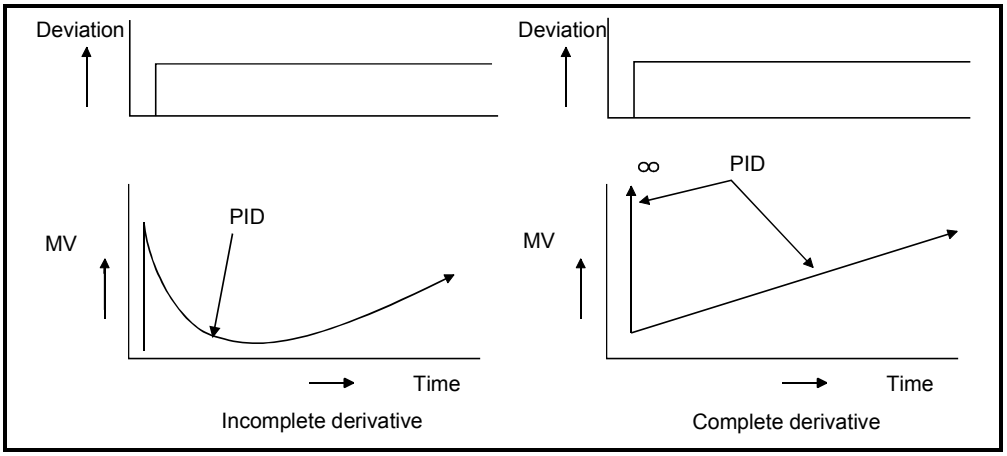


Fig. 4.5 PID Operation with Constant Deviation

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 mode to the automatic mode ..... The MV in the manual mode is transmitted to the MV area for the automatic mode.

(b) Changing from the automatic mode to the manual mode ..... The MV in the automatic mode is transmitted 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.

2) 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.

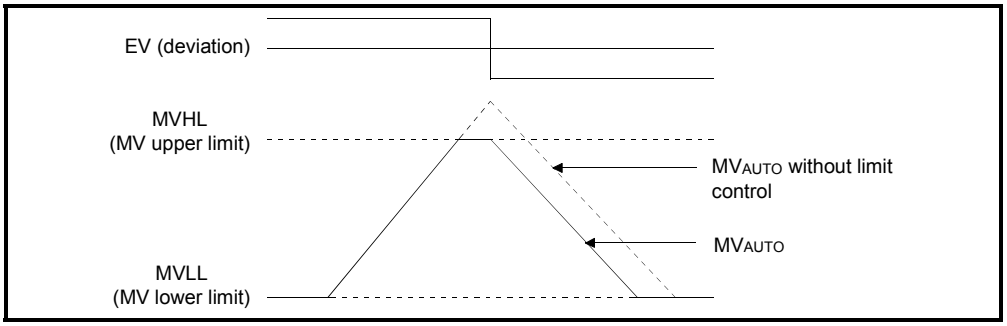


Fig. 4.6 Operation in Accordance with the MV Upper/Lower limit

(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 limit .....2000 (Or user-defined value)
- Lower limit .....0 (Or user-defined value)

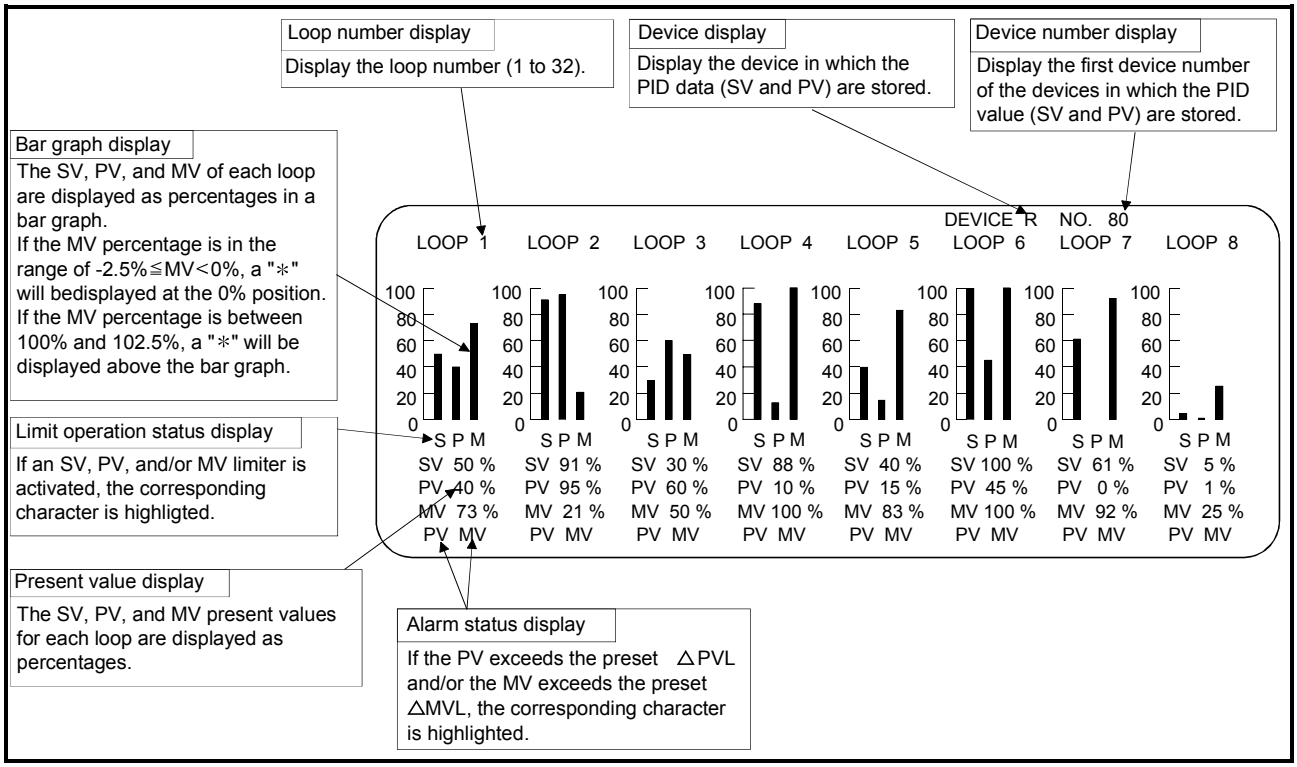
The value set for the upper limit must not be smaller than the value set for the lower limit.

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 ..... $\frac{PV}{2000} \times 100$ (%)

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		Operation
SM794 (Incomplete derivative)	SM774 (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

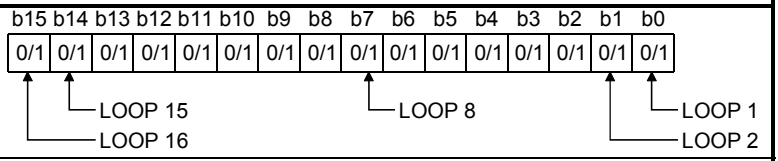
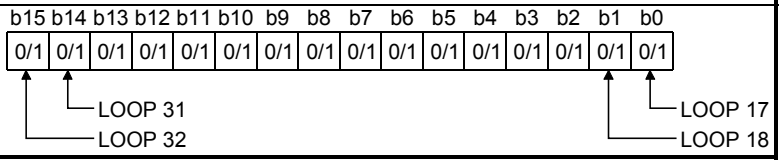
- The SV and PV are stored into the devices specified in the I/O data area with the PID operation instruction.

4.3.5 Changing the PID control data or input/output data setting range (for QCPU and LCPU)

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
PID control data	MV lower limit value
	MV upper limit value
	MV change rate limit value
	PV change rate limit value
I/O data	SV
	PV
	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		Setting Range
Incomplete derivative	Complete derivative	
SD794	SD774	
SD795	SD775	

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

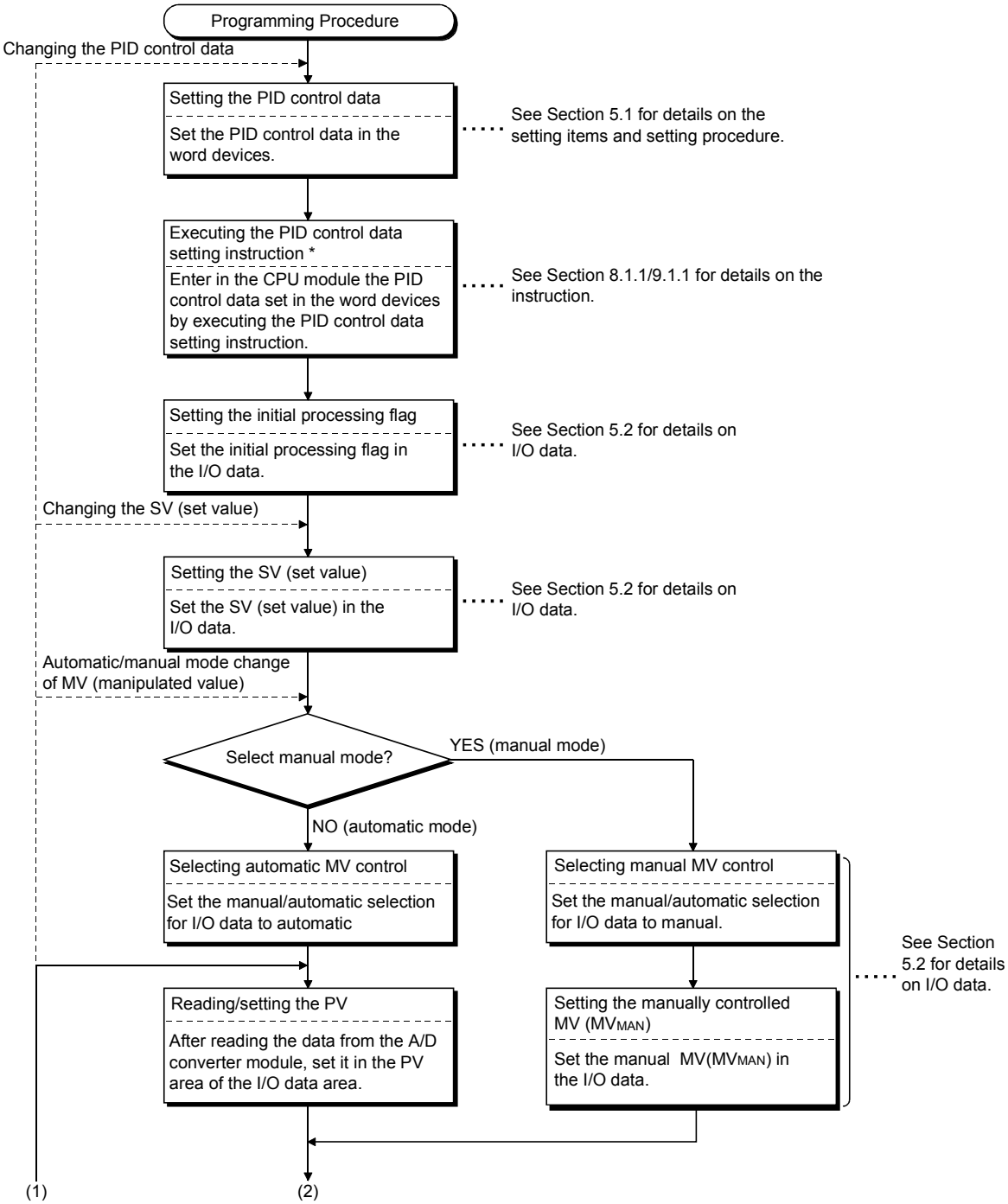
POINT

The Basic model QCPU has 8 loops.
b0 to b7 of SD774 and SD794 are valid.

MEMO

5. PID CONTROL PROCEDURE

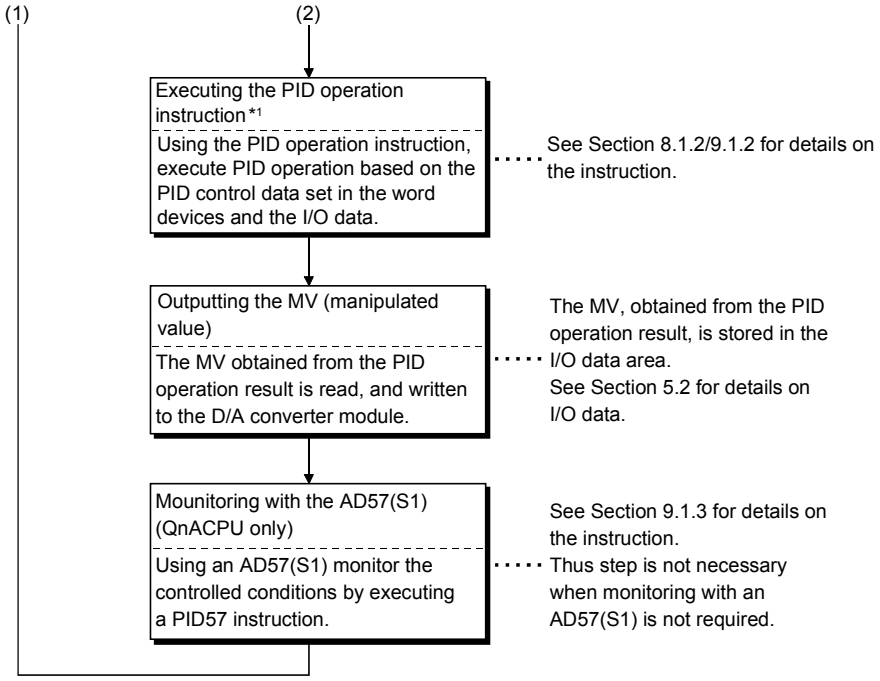
The programming procedure required to execute PID control is shown below.



REMARK

*: 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

Table 5.1 PID Control Data List

	Data No.	Data Item	Description	Incomplete derivative				
				With PID limits		Without PID limits		
				Setting Range	User Specification Range	Setting Range	User Specification Range	
Common setting data	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	
	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	
Data for each loop	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 (T_S)	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)	
	4	Integral constant (T_I)	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_I exceeds 3000.0 s	1 to 32767 (unit: 100 ms)	0.1 to 3000.0 s Infinite(∞) (If the setting for T_I exceeds 3000.0 s	1 to 32767 (unit: 100 ms)	
	5	Derivative constant (T_D)	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	

REMARK

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

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

	Complete derivative				Processing when Set Data is Outside the Allowable Setting Range
	With PID limits		Without PID limits		
	Setting Range	User Specification Range	Setting Range	User Specification Range	
	1 to 8	1 to 8	1 to 8	1 to 8	An error occurs and PID operation is not executed for all loops.
	1 to 8	1 to 8	1 to 8	1 to 8	
	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 corresponding loop is not executed.
	0.01 to 60.00s	1 to 6000 (unit: 10 ms)	0.01 to 60.00 s	1 to 6000 (unit: 10 ms)	
	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_I 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	

Table 5.1 PID Control Data List

	Data No.	Data Item	Description	Incomplete derivative				
				With PID limits		Without PID limits		
				Setting Range	User Specification Range	Setting Range	User Specification Range	
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	-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	
	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_D)	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.	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)	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)	

	Complete derivative				Processing when Set Data is Outside the Allowable Setting Range
	With PID limits		Without PID limits		
	Setting Range	User Specification Range	Setting Range	User Specification 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 less than -50, "-50" is used. ・When the MVLL or MVHL value is greater than 2050, "2050" is used.
	-50 to 2050	-50 to 2050	-32768 to 32767	-32768 to 32767	
	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, and LCPU

Table 5.2 PID Control Data List

	Data No.	Data Item	Description	Incomplete derivative				
				With PID limits		Without PID limits		
				Setting Range	User Specification Range	Setting Range	User Specification Range	
Common setting data	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	
	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	
Data for each loop	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 (T_S)	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)	
	4	Integral constant (T_I)	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_I exceeds 3000.0 s)	1 to 32767 (unit: 100 ms)	0.1 to 3000.0 s Infinite(∞) (If the setting for T_I exceeds 3000.0 s)	1 to 32767 (unit: 100 ms)	
	5	Derivative constant (T_D)	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	

	Complete derivative				Processing if Set Data is Outside the Allowable Setting Range
	With PID limits		Without PID limits		
	Setting Range	User Designation Range	Setting Range	User Designation Range	
	1 to 32	1 to 32	1 to 32	1 to 32	An error occurs and PID operation is not executed for all loops.
	1 to 32	1 to 32	1 to 32	1 to 32	
	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 corresponding loop is not executed.
	0.01 to 60.00 s	1 to 6000 (units: 10 ms)	0.01 to 60.00 s	1 to 6000 (units: 10 ms)	
	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 Infinite(∞) (If the setting for T_I exceeds 3000.0 s)	1 to 32767 (units: 100 ms)	0.1 to 3000.0 s Infinite(∞) (If the setting for T_I 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	

Table 5.2 PID Control Data List

	Data No.	Data Item	Description	Incomplete derivative				
				With PID limits		Without PID limits		
				Setting Range	User Specification Range	Setting Range	User Specification Range	
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	-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	
	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_D)	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.	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)	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)	

	Complete derivative				Processing when Set Data is Outside the Allowable Setting Range
	With PID limits		Without PID limits		
	Setting Range	User Specification Range	Setting Range	User Specification 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. ・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.
	-50 to 2050	-50 to 2050	-32768 to 32767	-32768 to 32767	
	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

	Data No.	Data Item	Description	Setting Range	User Specification Range	Processing when Set Data is Outside the Allowable Setting Range
Common setting data	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 operation is not executed for all loops.
	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	
Data for each loop	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 loop is not executed.
	2	Sampling cycle (T_S)	Sets the cycle of PID operation.	0.01 to 60.00 s	1 to 6000 (unit: 10 ms)	
	3	Proportional constant (K_P)	PID operation ratio	0.01 to 100.00	1 to 10000 (unit: 0.01)	
	4	Integral constant (T_I)	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_I exceeds 3000.0 s	1 to 32767 (unit: 100 ms)	An error occurs and PID operation for the corresponding loop is not executed.
	5	Derivative constant (T_D)	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 loop is not executed.
	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	
	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 value. ・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.
	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	
	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	Conmon to all loops
+1	Number of loops in 1 scan	
+2	Selection of operational expression	For loop No.1 (14 words)
+3	Sampling cycle (T_S)	
+4	Proportional constant (K_P)	
+5	Integral constant (T_I)	
+6	Derivative constant (T_D)	
+7	Filter coefficient (α)	
+8	MV lower limit (MVLL)	
+9	MV upper limit (MVHL)	
+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	For loop No.2 (14 words)
+17	Sampling cycle (T_S)	
+18	Proportional constant (K_P)	
+19	Integral constant (T_I)	
+20	Derivative constant (T_D)	
+21	Filter coefficient (α)	
+22	MV lower limit (MVLL)	
+23	MV upper limit (MVHL)	
+24	MV change rate limit (△MVL)	
+25	PV change rate limit (△PVL)	
+26	0 *	
+27	Derivative gain (K_D)	
+28	0 *	
+29	0 *	
to	to	
+(m + 0)	Selection of operational expression	For loop No.n (14 words)
+(m + 1)	Sampling cycle (T_S)	
+(m + 2)	Proportional constant (K_P)	
+(m + 3)	Integral constant (T_I)	
+(m + 4)	Derivative constant (T_D)	
+(m + 5)	Filter coefficient (α)	
+(m + 6)	MV lower limit (MVLL)	
+(m + 7)	MV upper limit (MVHL)	
+(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 *	

For the total number of loops to be used

$m=(n-1) \times 14+2$

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)

(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

Specified device number +0	Number of loops	Conmon to all loops
+1	Number of loops in 1 scan	
+2	Selection of operational expression	For loop No.1 (10 words)
+3	Sampling cycle (T_S)	
+4	Proportional constant (K_P)	
+5	Integral constant (T_I)	
+6	Derivative constant (T_D)	
+7	Filter coefficient (α)	
+8	MV lower limit (MVLL)	
+9	MV upper limit (MVHL)	
+10	MV change rate limit (△MVL)	
+11	PV change rate limit (△PVL)	
+12	Selection of operational expression	For loop No.2 (10 words)
+13	Sampling cycle (T_S)	
+14	Proportional constant (K_P)	
+15	Integral constant (T_I)	
+16	Derivative constant (T_D)	
+17	Filter coefficient (α)	
+18	MV lower limit (MVLL)	
+19	MV upper limit (MVHL)	
+20	MV change rate limit (△MVL)	
+21	PV change rate limit (△PVL)	
+22	Selection of opperational expression	For loop No.3 (10 words)
+23	Sampling cycle (T_S)	
to	to	
+(m + 0)	Selection of opperational expression	For loop No.n (10 words)
+(m + 1)	Sampling cycle (T_S)	
+(m + 2)	Proportional constant (K_P)	
+(m + 3)	Integral constant (T_I)	
+(m + 4)	Derivative constant (T_D)	
+(m + 5)	Filter coefficient (α)	
+(m + 6)	MV lower limit (MVLL)	
+(m + 7)	MV upper limit (MVHL)	
+(m + 8)	MV change rate limit (△MVL)	
+(m + 9)	PV change rate limit (△PVL)	

For the total number of loops to be used

m=(n-1) × 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 × 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

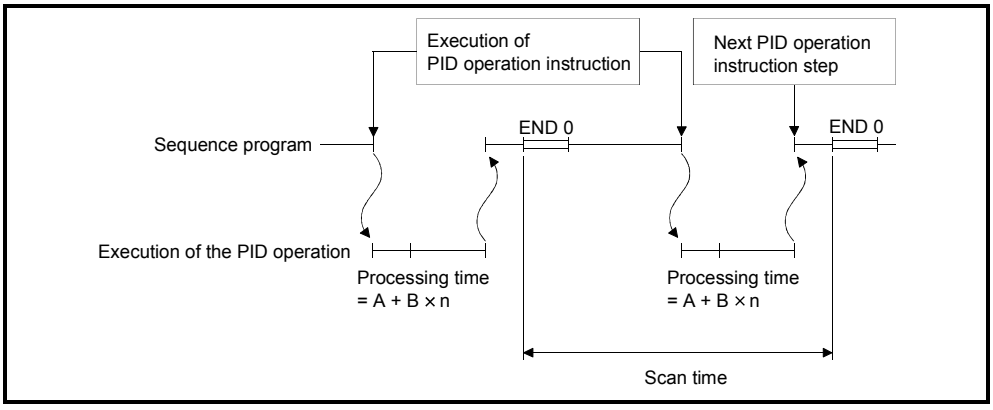
(1) The number of loops to be used means the number of loops for which PID operation is executed. The sampling cycle time is measured for the set number of loops when the PID operation instruction* is executed. PID operation is executed for the loop for which the sampling cycle time reaches or exceeds 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 × 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

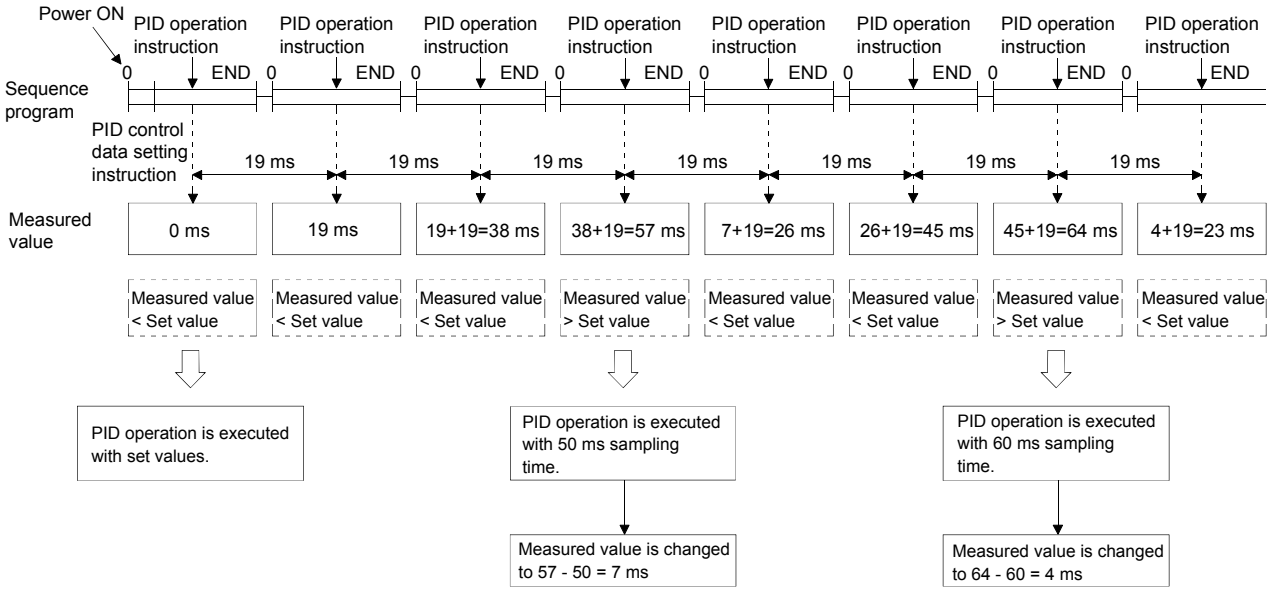
*: The following instructions are available as the PID operation instructions.
- S.PIDCONT (incomplete derivative)
- PIDCONT (complete derivative)

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.

When sampling cycle = 50 ms



POINT

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)

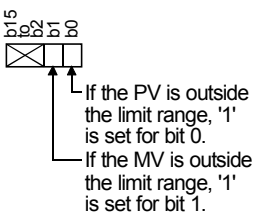
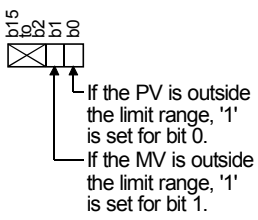
MEMO

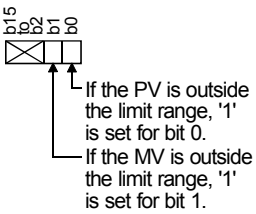
5.2 I/O Data

(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-by-loop" and "work area used by the system to perform PID operation".

Table 5.4 I/O Data List

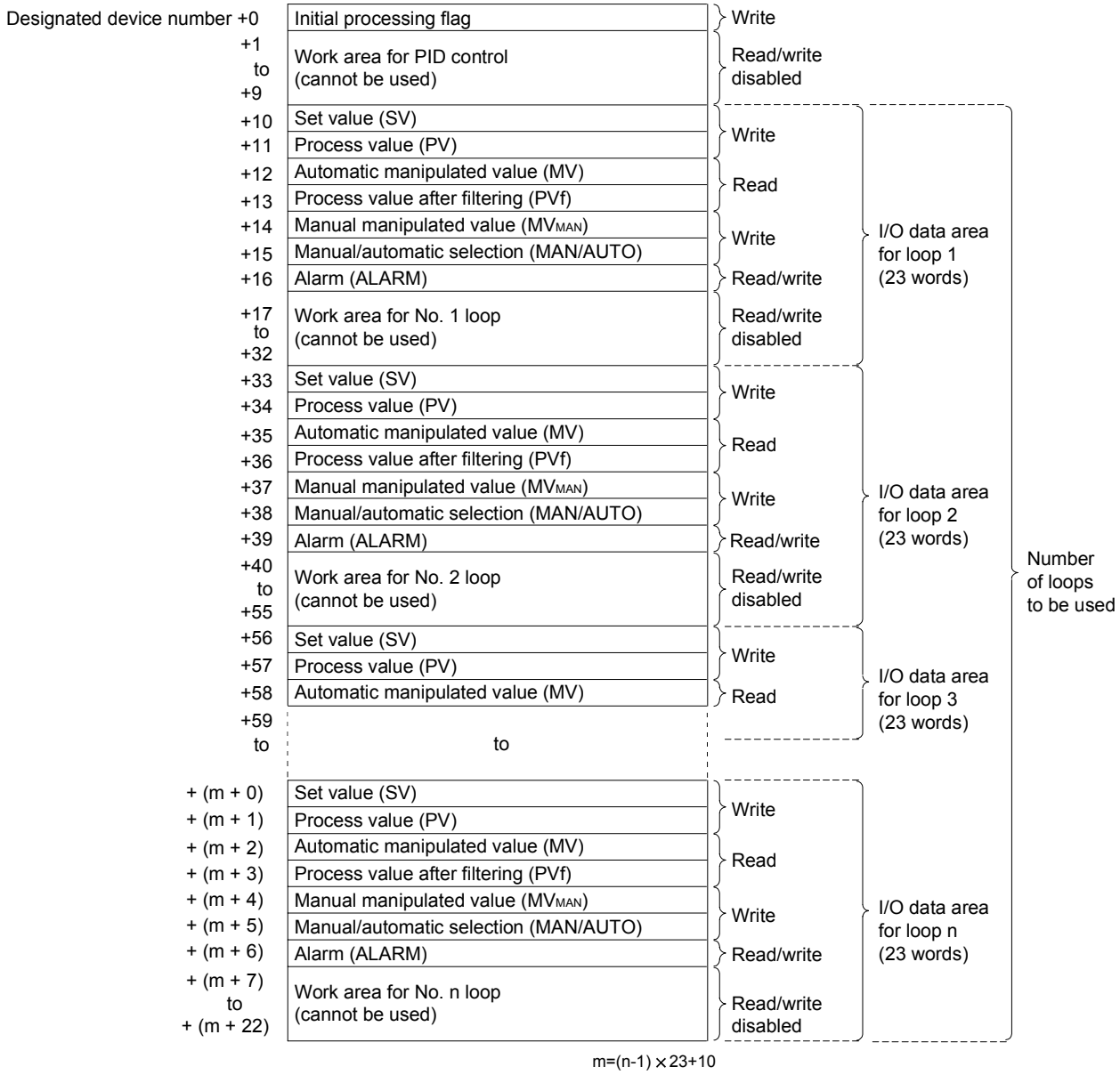
Data Name		Description	Setting Range		
			QCPU, LCPU		
			With PID limits	Without PID limits	
Set value	SV	・PID control target value	0 to 2000	-32768 to 32767	
Process value	PV	・Feedback data from controlled system objective to A/D converter module	-50 to 2050	-32768 to 32767	
Automatic manipulated value	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	MV_{MAN}	・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).			

	QnACPU	Processing when Set Data is Outside the Specified Range
	0 to 2000	For the QCPU or LCPU with "With PID limits" setting or QnACPU, convert it to the following value for PID operation. ・When SV is less than 0, SV must be 0. ・When SV is greater than 2000, SV must be 2000.
	-50 to 2050	For the QCPU or LCPU with "With PID limits" setting or QnACPU, convert it to the following value for PID operation. ・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	For the QCPU or LCPU with "With PID limits" setting or QnACPU, convert it to the following value for PID operation. ・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.
		—

(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



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)

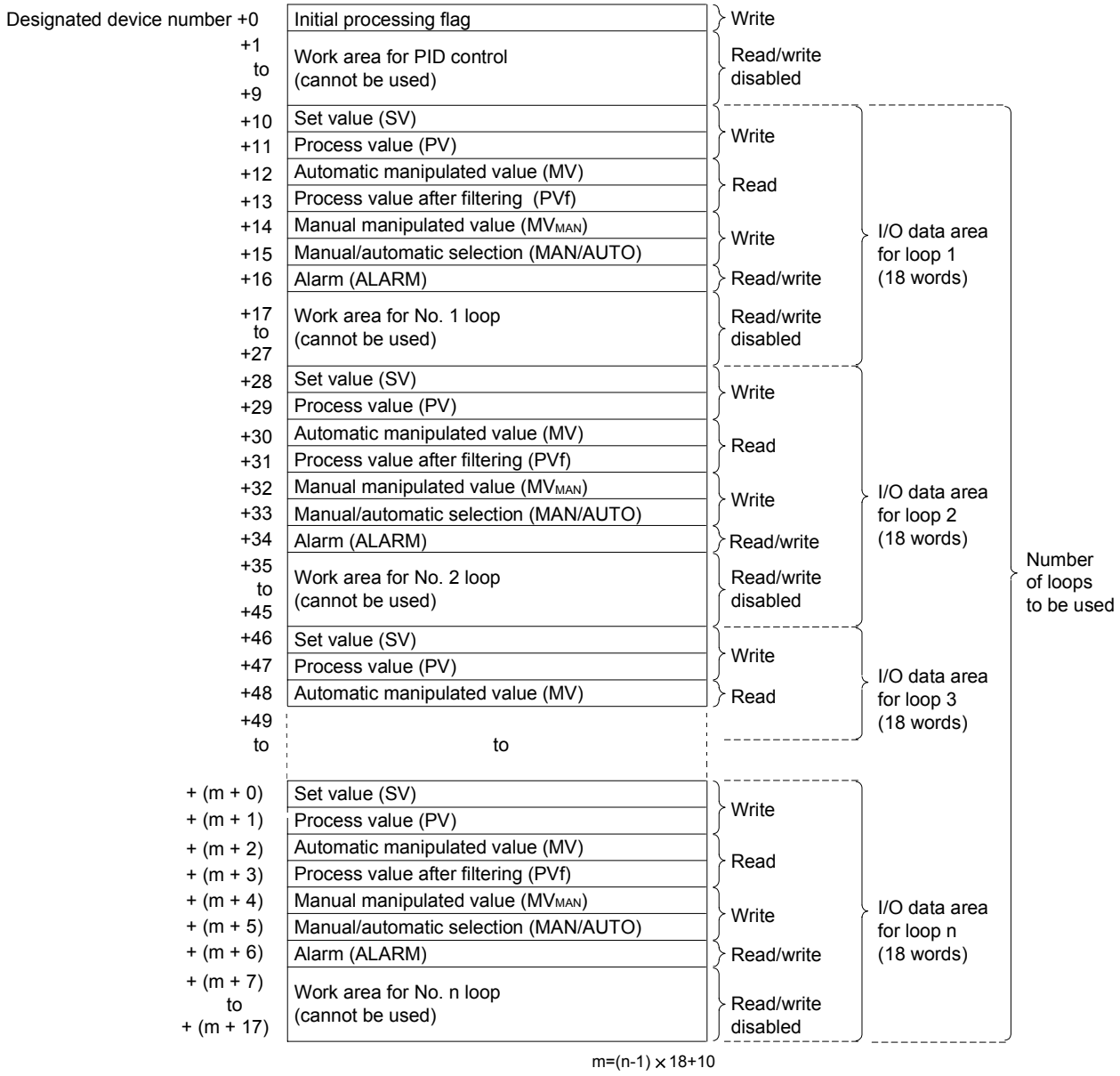
2) Set each data as a binary value.

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) 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 × 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.
   Note that when control is to be started from the initial status, data must be cleared with a sequence program.

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)

MEMO

6. PID CONTROL INSTRUCTIONS

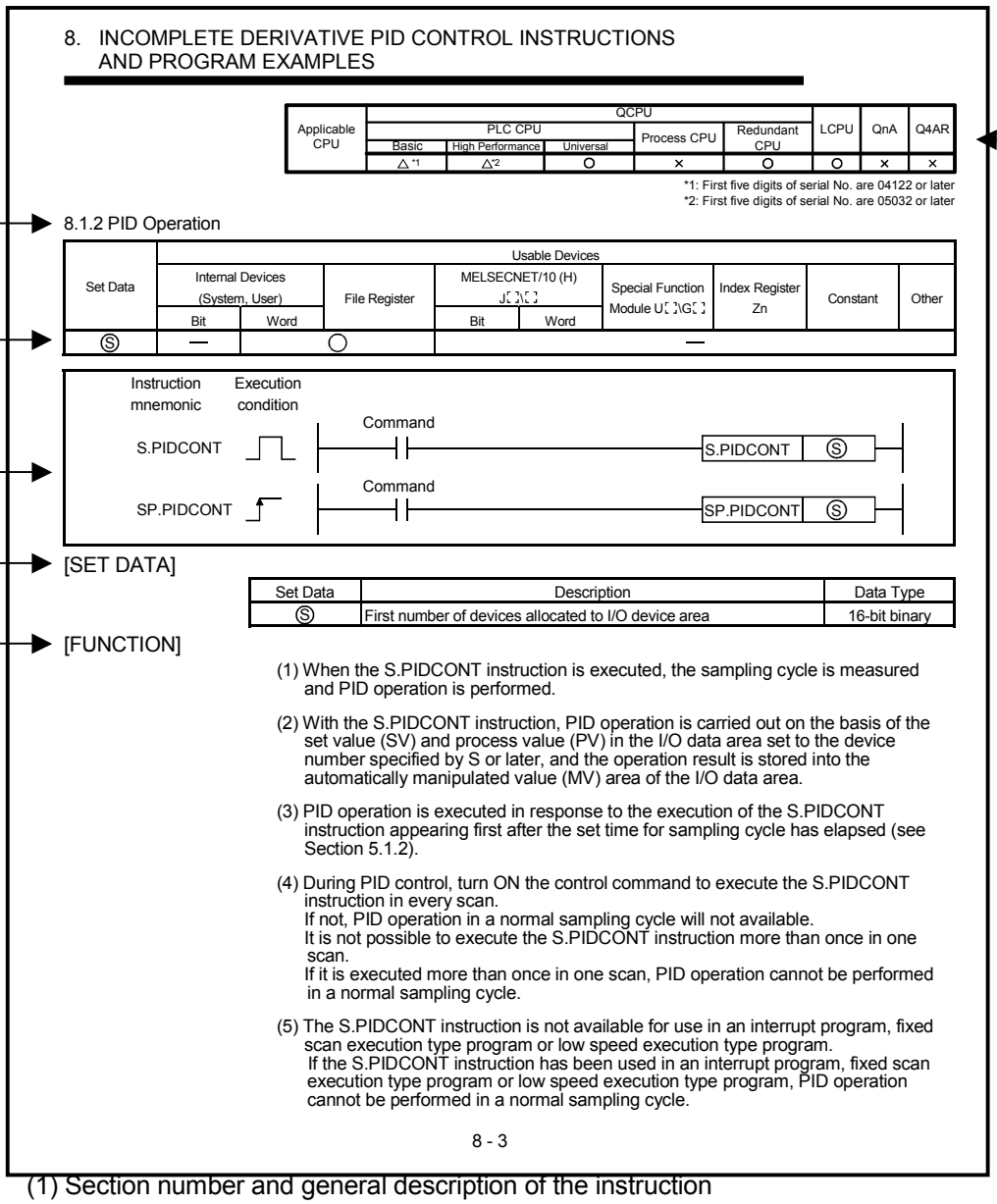
The structure of PID control instructions is the same as that of common instructions for the Basic model QCPU, High Performance model QCPU, Redundant CPU, Universal model QCPU, LCPU, and QnACPU.

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.



(1) Section number and general description of the instruction
(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 follows.

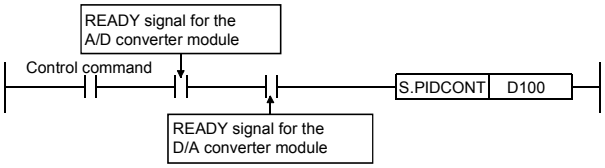
Device Classification	Internal Device (system, user)		File Register	MELSECNET/10 (H) J□\□*3		Special Function Module U□\G□	Index Register Zn	Constant*1	Other*1
	Bit	Word		Bit	Word				
Usable devices*4	X, Y, M, L, SM, F, B, SB, FX*2, FY*2	T*5, ST*5, C*5,D, W, SD,SW, FD	R, ZR	J□\X J□\Y J□\B J□\SB	J□\W J□\SW	U□\G□	Z	Decimal number Hexadecimal number Real number constant Character string constant	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 explanation of each device, refer to the User's Manual (Function Explanation, Program Fundamentals) of the CPU module used, or the QnACPU Programming Manual (Fundamentals).
*5 : T, ST and C can be used as the word devices only.

(6) For Ⓢ, 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.

(7) → [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 Ⓢ, 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.

(3) Indicates the expressions and instruction execution conditions in the ladder mode.

Execution Condition	Executed while ON	Executed once at OFF → ON
Symbol used on the explanation page		

(4) Explains the set data for each instruction and indicates the data type.

Data Type	Description
16-bit binary	Indicates that binary 16-bit data or the first number of a word device can be used.

(5) Indicates the function of the instruction.

(6) This shows a CPU module to which instructions are given.
○: Usable, △: Usable on condition, ×: Unusable

(7) Indicates the conditions that will cause errors and the error numbers.

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

Applicable CPU	QCPU					LCPU	QnA	Q4AR
	PLC CPU			Process CPU	Redundant CPU			
	Basic	High Performance	Universal					
	△ *1	△*2	○	×	○	○	×	×

*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

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

[SET DATA]

Set Data	Description	Data Type
Ⓢ	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 Ⓢ 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

(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 Ⓢ, 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)

Applicable CPU	QCPU					LCPU	QnA	Q4AR
	PLC CPU			Process CPU	Redundant CPU			
	Basic	High Performance	Universal					
	△ *1	△*2	○	×	○	○	×	×

*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

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

[SET DATA]

Set Data	Description	Data Type
Ⓢ	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).

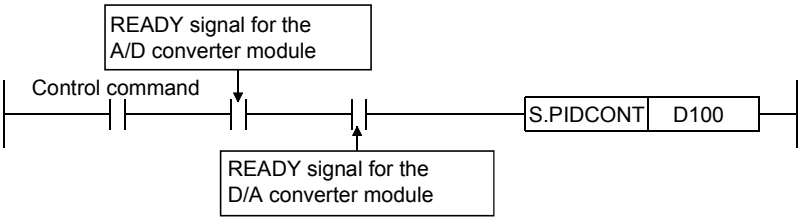
(4) During PID control, turn ON the control command to execute the S.PIDCONT instruction in every scan.
If not, PID operation in a normal sampling cycle will not available.
It is not possible to execute the S.PIDCONT instruction more than once in one scan.
If it is executed more than once in one scan, 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 Ⓢ, 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 (MV_{MAN}) 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 not 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 Ⓢ, 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.

Applicable CPU	QCPU					LCPU	QnA	Q4AR
	PLC CPU			Process CPU	Redundant CPU			
	Basic	High Performance	Universal					
	△ *1	△*2	○	×	○	○	×	×

*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.

Set Data	Usable Devices							
	Internal Devices (System, User)		File Register	MELSECNET/10 (H) Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
ⓝ	○			—				○	—

[SET DATA]

Set Data	Description	Data Type
ⓝ	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 ⓝ.
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 ⓝ.
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.

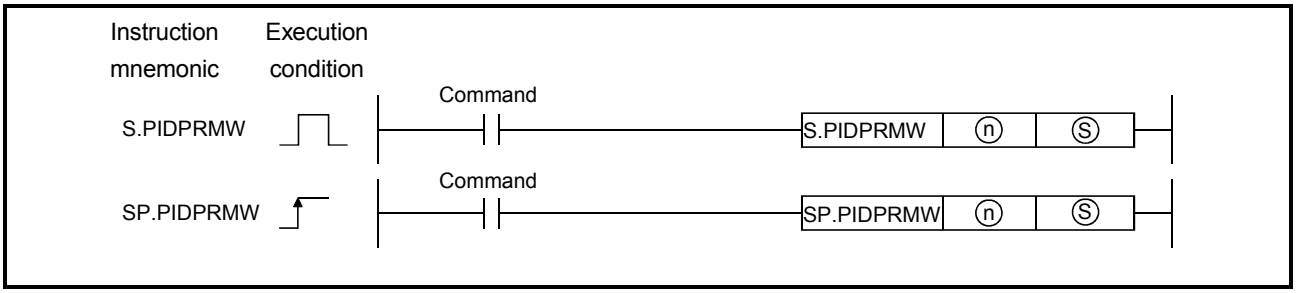
- When the loop number designated by ⓝ does not exist. (Error code: 4100)
- When ⓝ is outside the range 1 to 8. (Basic model QCPU) (Error code: 4100)
- When ⓝ is any other than 1 to 32. (High Performance model QCPU, Redundant CPU, Universal model QCPU, LCPU) (Error code: 4100)
- When the S.PIDINIT and 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)

Applicable CPU	QCPU					LCPU	QnA	Q4AR
	PLC CPU			Process CPU	Redundant CPU			
	Basic	High Performance	Universal					
	△ *1	△*2	○	×	○	○	×	×

*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

Set Data	Usable Devices							
	Internal Devices (System, User)		File Register	MELSECNET/10 (H) Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
ⓝ	○	○		—				○	—
Ⓢ	—	○		—				—	—



[SET DATA]

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

[FUNCTIONS]

(1) Changes the operation parameter for the loop number designated by ⓝ to the PID control data stored in the devices starting with the device number designated by Ⓢ.

(2) The configuration of the data for PID control which starts from the device number designated by Ⓢ is shown below.For details on PID control data, see Section 5.1.

Ⓢ +0	Selection of operational expression
Ⓢ +1	Sampling cycle (T_S)
Ⓢ +2	Proportional constant (K_P)
Ⓢ +3	Integral constant (T_I)
Ⓢ +4	Derivative constant (T_D)
Ⓢ +5	Filter coefficient (α)
Ⓢ +6	MV lower limit (MVLL)
Ⓢ +7	MV upper limit (MVHL)
Ⓢ +8	MV change rate limit (△MVL)
Ⓢ +9	PV change rate limit (△PVL)
Ⓢ +10	0
Ⓢ +11	Derivative gain (K_D)
Ⓢ +12	0
Ⓢ +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.

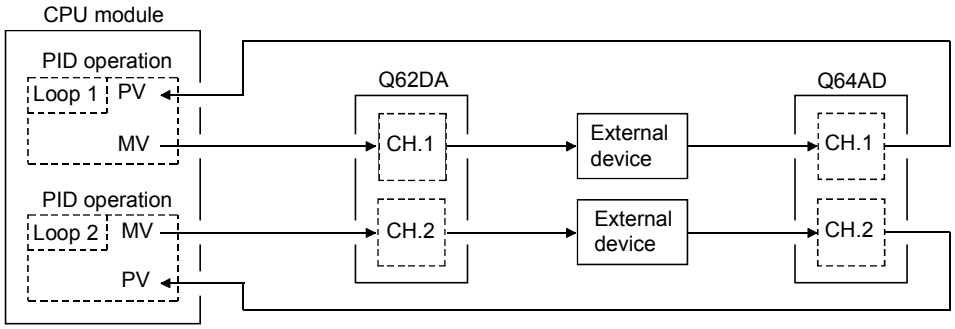
- When the loop number designated by ⓝ does not exist. (Error code: 4100)
- When ⓝ is outside the range 1 to 8.(Basic model QCPU) (Error code: 4100)
- When ⓝ is any other than 1 to 32. (High Performance model QCPU, Redundant CPU, Universal model QCPU, LCPU) (Error code: 4100)
- When the PID control data is outside the setting range. (Error code: 4100)
- When any of Ⓢ+10, Ⓢ+12 and Ⓢ+13 in the PID control data is not 0. (Error code: 4100)
- When the device range assigned to the PID control data area by Ⓢ 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: 4103)

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 numbers .....X/Y80 to X/Y8F
Q62DA I/O numbers .....X/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.

(4) The PID control data is set in the following devices:*1
Common data .....D500 and D501
Loop 1 data .....D502 to D515
Loop 2 data .....D516 to D529

(5) The I/O data is set in the following devices:*2
Common data .....D600 to D609
Loop 1 data .....D610 to D632
Loop 2 data .....D633 to D655

(6) The following SV are set for loop 1 and loop 2 using a sequence program:
Loop 1 600
Loop 2 1000

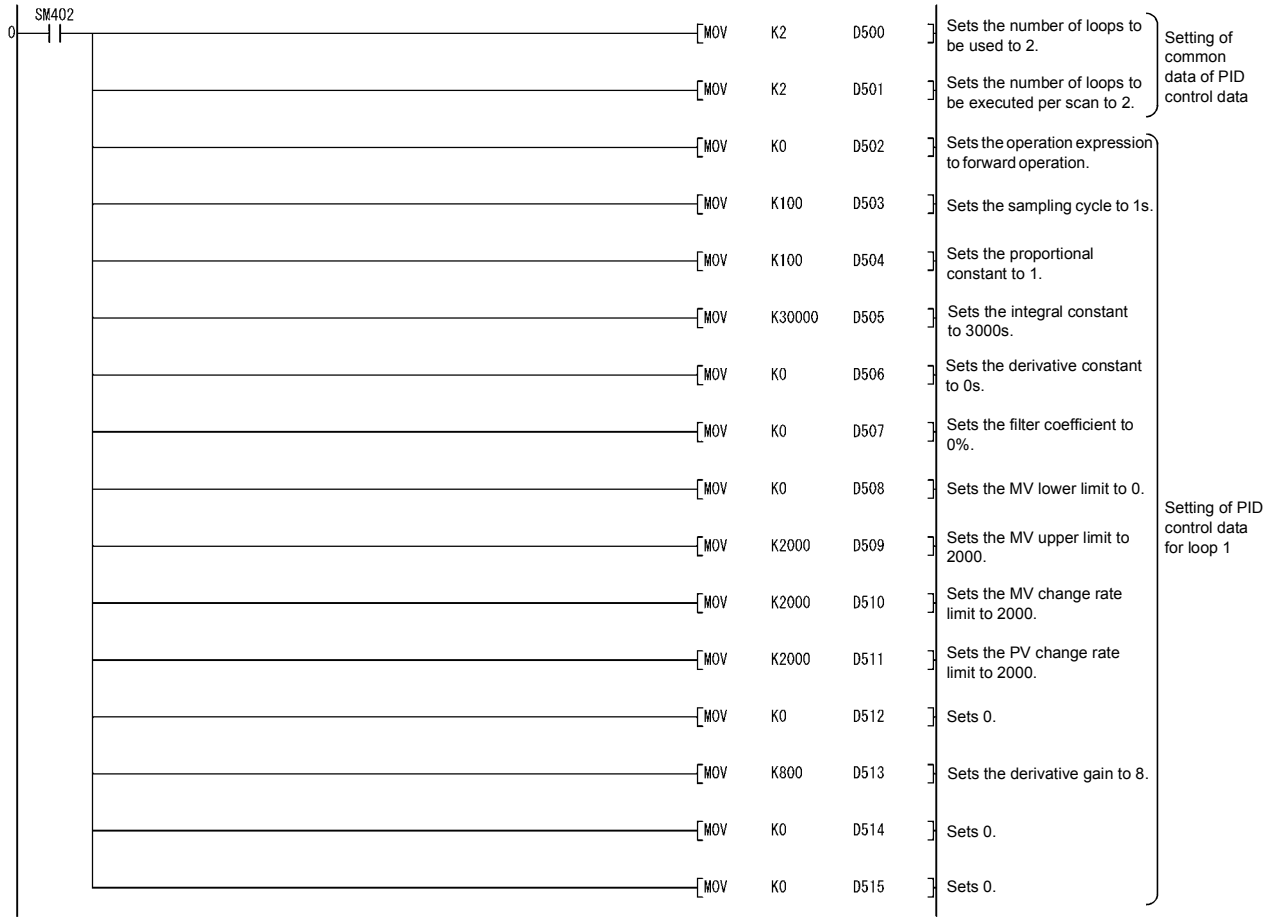
(7) The following devices are used for PID control start/stop commands.
PID control start command X0
PID control stop command X1

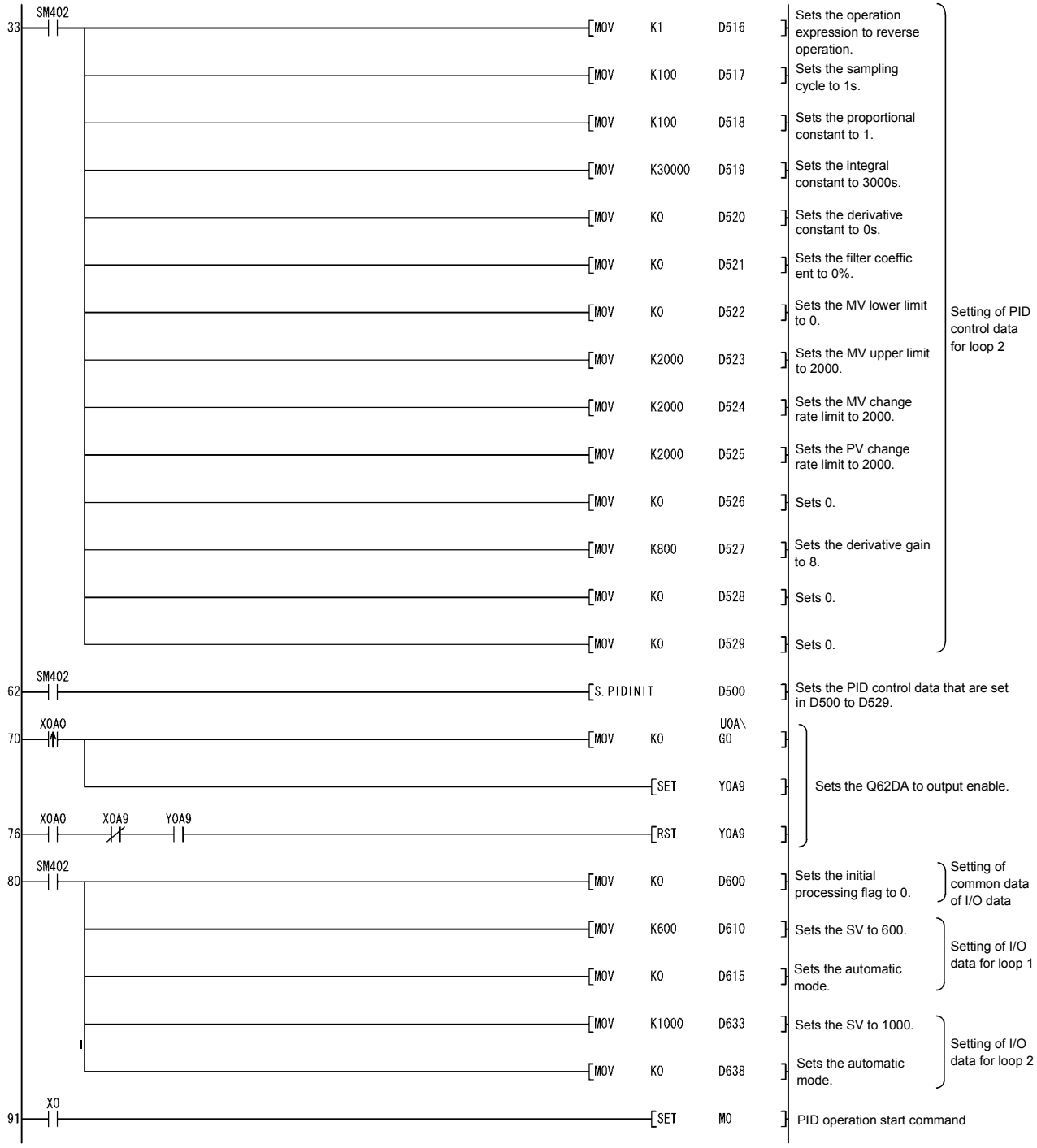
(8) The digital values of the Q64AD and Q62DA are set within the range 0 to 2000.

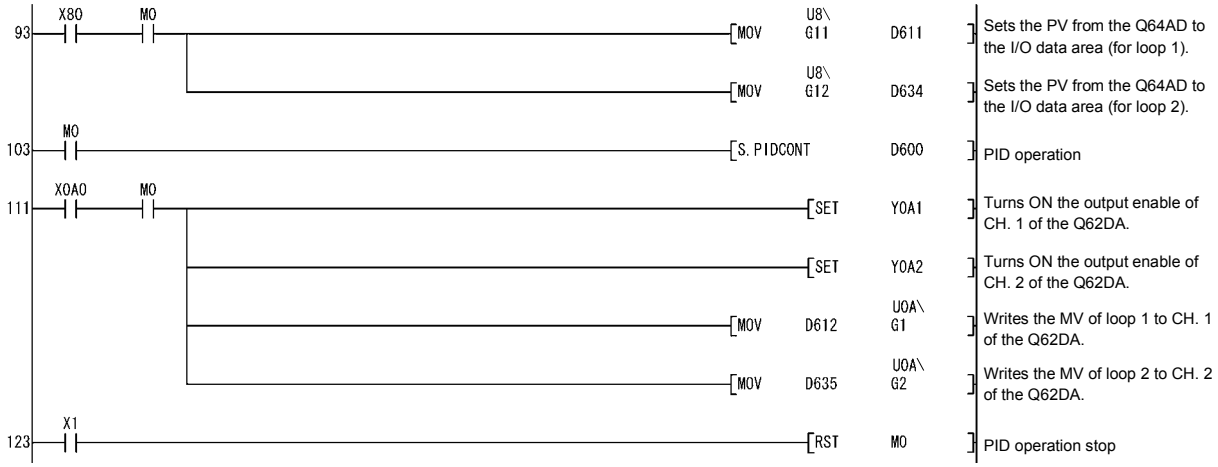
REMARK

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

[PROGRAM EXAMPLE]




33 SM402
MOV K1 D516 Sets the operation expression to reverse operation.
MOV K100 D517 Sets the sampling cycle to 1s.
MOV K100 D518 Sets the proportional constant to 1.
MOV K30000 D519 Sets the integral constant to 3000s.
MOV K0 D520 Sets the derivative constant to 0s.
MOV K0 D521 Sets the filter coeffic ent to 0%.
MOV K0 D522 Sets the MV lower limit to 0.
MOV K2000 D523 Sets the MV upper limit to 2000.
MOV K2000 D524 Sets the MV change rate limit to 2000.
MOV K2000 D525 Sets the PV change rate limit to 2000.
MOV K0 D526 Sets 0.
MOV K800 D527 Sets the derivative gain to 8.
MOV K0 D528 Sets 0.
MOV K0 D529 Sets 0.
Setting of PID control data for loop 2
62 SM402
S.PIDINIT D500 Sets the PID control data that are set in D500 to D529.
70 XOA0
MOV K0 UOA\G0
SET YOA9
Sets the Q62DA to output enable.
76 XOA0 XOA9 YOA9
RST YOA9
80 SM402
MOV K0 D600 Sets the initial processing flag to 0.
Setting of common data of I/O data
MOV K600 D610 Sets the SV to 600.
MOV K0 D615 Sets the automatic mode.
Setting of I/O data for loop 1
MOV K1000 D633 Sets the SV to 1000.
MOV K0 D638 Sets the automatic mode.
Setting of I/O data for loop 2
91 X0
SET M0 PID operation start command


93
X80
M0
MOV
U8\G11
D611
Sets the PV from the Q64AD to the I/O data area (for loop 1).
MOV
U8\G12
D634
Sets the PV from the Q64AD to the I/O data area (for loop 2).
103
M0
S.PIDCONT
D600
PID operation
111
X0A0
M0
SET
Y0A1
Turns ON the output enable of CH. 1 of the Q62DA.
SET
Y0A2
Turns ON the output enable of CH. 2 of the Q62DA.
MOV
D612
U0A\G1
Writes the MV of loop 1 to CH. 1 of the Q62DA.
MOV
D635
U0A\G2
Writes the MV of loop 2 to CH. 2 of the Q62DA.
123
X1
RST
M0
PID operation stop

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.

(4) The PID control data is set in the following devices:
Common data ..... D500 and D501
Loop 1 data ..... D502 to D515

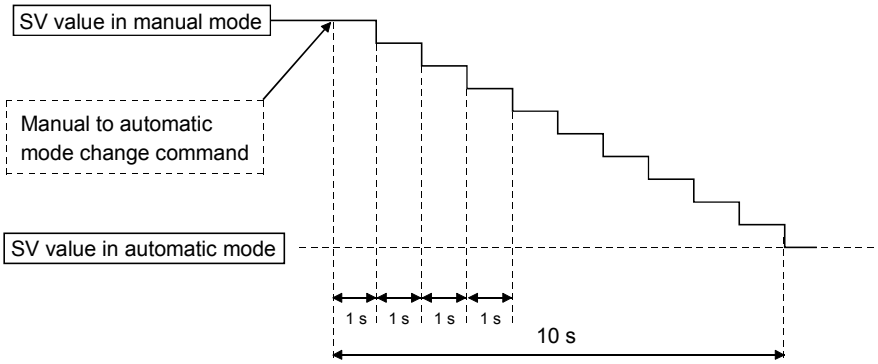
(5) The I/O data is set in the following devices:
Common data ..... D600 to D609
Loop 1 data ..... D610 to D630

(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

(7) The following devices are used to start and stop PID control and the automatic/manual changeover command:
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 command 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, 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:

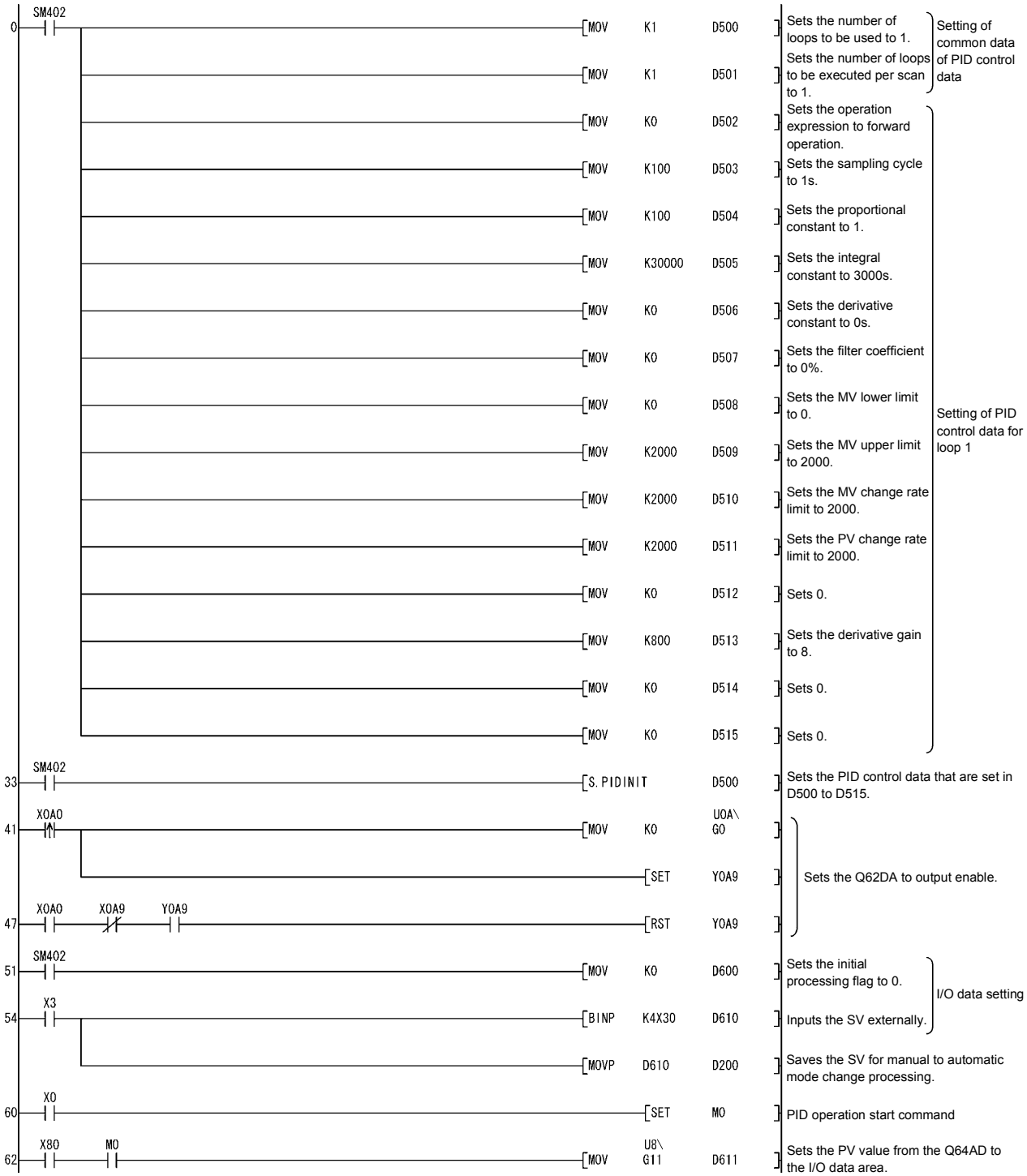


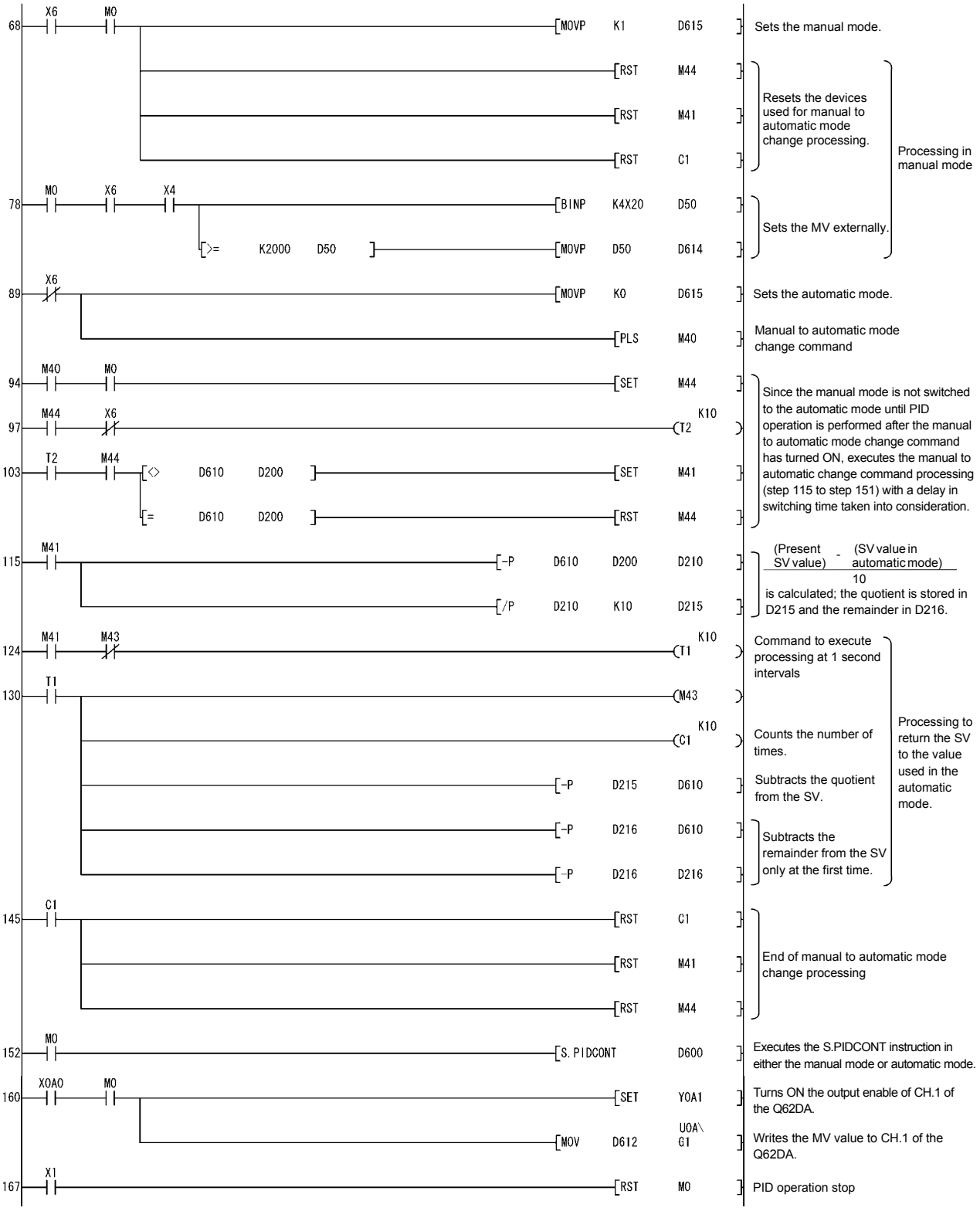
The SV is rewritten using the operation method illustrated below:

$$\frac{(\text{SV value in manual mode}) - (\text{SV value in automatice mode})}{10} = \boxed{\text{Incremental value}} \cdots\cdots \boxed{\text{Remainder}}$$

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

[PROGRAM EXAMPLE]




68
X6
M0
MOVP K1 D615
Sets the manual mode.
RST M44
RST M41
RST C1
Resets the devices used for manual to automatic mode change processing.
Processing in manual mode
78
M0
X6
X4
BINP K4X20 D50
>= K2000 D50
MOVP D50 D614
Sets the MV externally.
89
X6
MOVP K0 D615
Sets the automatic mode.
PLS M40
Manual to automatic mode change command
94
M40
M0
SET M44
97
M44
X6
K10
T2
103
T2
M44
<> D610 D200
SET M41
= D610 D200
RST M44
Since the manual mode is not switched to the automatic mode until PID operation is performed after the manual to automatic mode change command has turned ON, executes the manual to automatic change command processing (step 115 to step 151) with a delay in switching time taken into consideration.
115
M41
-P D610 D200 D210
/P D210 K10 D215
(Present SV value) - (SV value in automatic mode) / 10
is calculated; the quotient is stored in D215 and the remainder in D216.
124
M41
M43
K10
T1
Command to execute processing at 1 second intervals
130
T1
M43
K10
C1
Counts the number of times.
-P D215 D610
Subtracts the quotient from the SV.
-P D216 D610
Subtracts the remainder from the SV only at the first time.
-P D216 D216
Processing to return the SV to the value used in the automatic mode.
145
C1
RST C1
RST M41
RST M44
End of manual to automatic mode change processing
152
M0
S.PIDCONT D600
Executes the S.PIDCONT instruction in either the manual mode or automatic mode.
160
X0A0
M0
SET Y0A1
Turns ON the output enable of CH.1 of the Q62DA.
MOV D612 U0\G1
Writes the MV value to CH.1 of the Q62DA.
167
X1
RST M0
PID operation stop

9. COMPLETE DERIVATIVE PID CONTROL INSTRUCTIONS AND PROGRAM EXAMPLES

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

9.1 PID Control Instructions

Applicable CPU	QCPU					LCPU	QnA	Q4AR
	PLC CPU			Process CPU	Redundant CPU			
	Basic	High Performance	Universal					
	△ *1	○	○	×	○	○	○	○

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

9.1.1 PID control data settings

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

Instruction mnemonic	Execution condition	
PIDINIT	Command	PIDINIT Ⓢ
PIDINITP	Command	PIDINITP Ⓢ

[SET DATA]

Set Data	Description	Data Type
Ⓢ	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 Ⓢ 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]

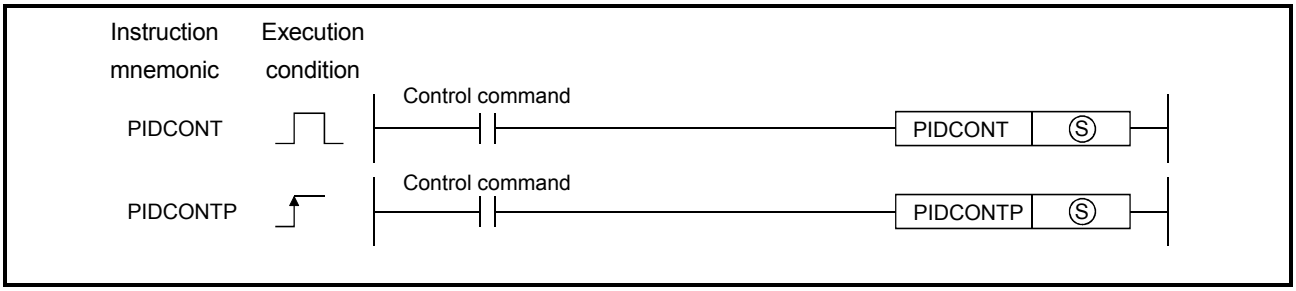
(1) An operation error will occur, the error flag (SM0) will be turned ON, and an error code will be stored in SD0, in the following cases.
- When the value set as the PID control data is outside the allowable range. (Error code: 4100)
- When (Number of loops used) < (Number of loops executed in one 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 Ⓢ, exceeds the last device number of the corresponding device. (Error code: 4101)

Applicable CPU	QCPU					LCPU	QnA	Q4AR
	PLC CPU			Process CPU	Redundant CPU			
	Basic	High Performance	Universal					
	△ *1	○	○	×	○	○	○	○

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

9.1.2 PID control

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



[SET DATA]

Set Data	Description	Data Type
Ⓢ	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).

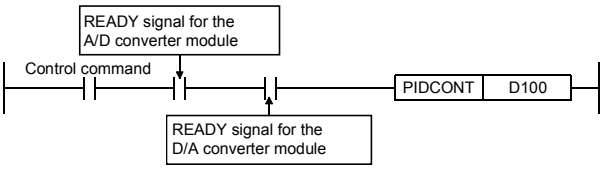
(4) During PID control, turn ON the control command to execute the PIDCONT instruction in every scan.
If not, PID operation in a normal sampling cycle will not available.
It is not possible to execute the PIDCONT instruction more than once in one scan.
If it is executed more than once in one scan, PID 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 Ⓢ, 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 not 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 Ⓢ, exceeds the last device number of the corresponding device. (Error code: 4101)

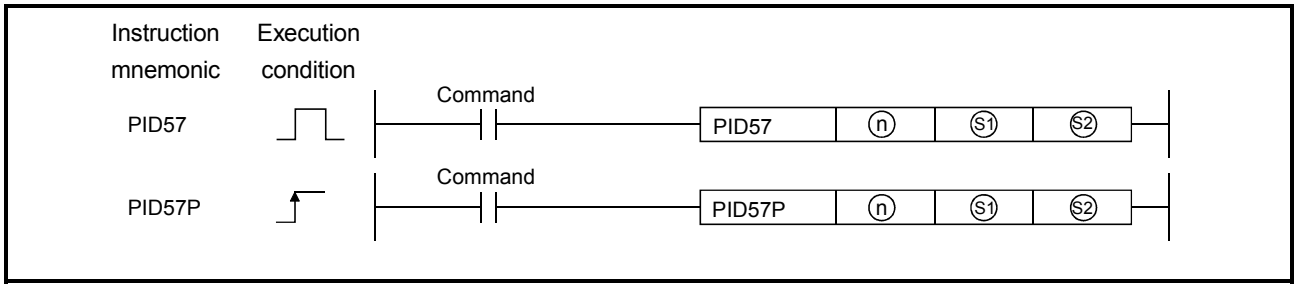
REMARK

*: 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.

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

9.1.3 Monitoring PID control status (QnACPU only)

Set Data	Usable Devices							
	Internal Devices (System, User)		File Register	MELSECNET/10 (H) Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
ⓝ	○							○	—
ⓢ1	○							○	—
ⓢ2	○							—	—



[SET DATA]

Set Data	Description	Data Type
ⓝ	First I/O number of the AD57(S1) used to monitor the PID control status	16-bit binary
ⓢ1	Screen number corresponding to the loop number to be monitored	
ⓢ2	Initial screen display request	

[FUNCTION]

(1) The display unit of the AD57(S1) designated by ⓝ displays the PID control status of the loop number designated by ⓢ1 in a bar graph.
By executing the initial screen display request, designated by ⓢ2, 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 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 (S1) 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 (S2), 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 (S1) is automatically stored in (S2) and then the PID control monitor function is executed.
If the device designated by (S2) 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 correctly.

(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 00B$_H$, 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

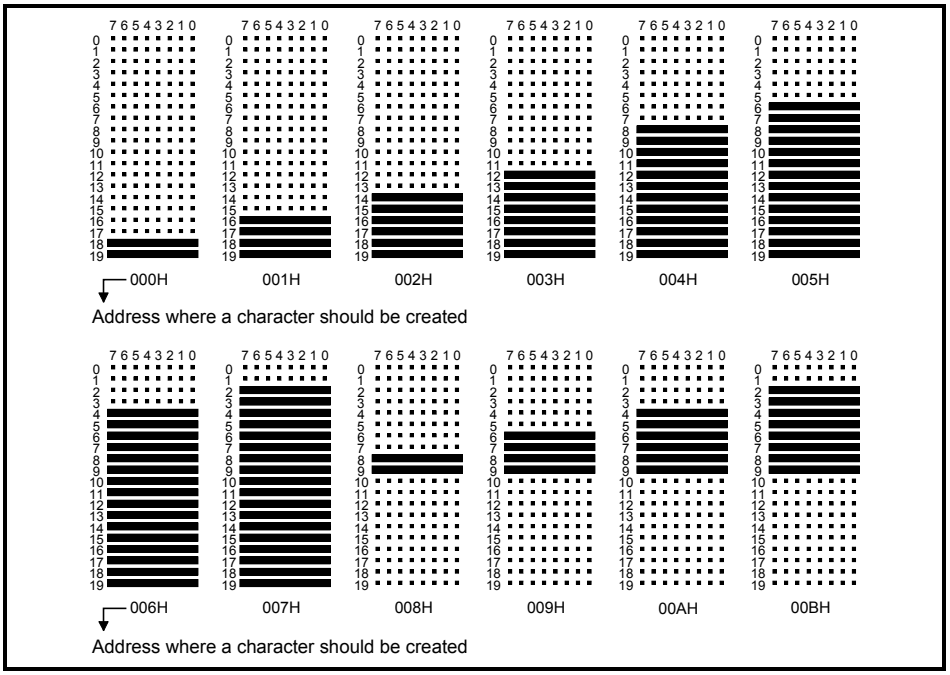


Fig. 9.1 Characters for PID Control Status Monitor

[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 (S1) is outside the range of 1 to 4. (Error code: 4100)

Applicable CPU	QCPU					LCPU	QnA	Q4AR
	PLC CPU			Process CPU	Redundant CPU			
	Basic	High Performance	Universal					
	△*1	○	○	×	○	○	○	○

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

9.1.4 Operation stop/start of designated loop no.

Set Data	Usable Devices							
	Internal Devices (System, User)		File Register	MELSECNET/10 (H) Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
ⓝ	○								—

[SET DATA]

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

[FUNCTION]

(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.

[OPERATION ERRORS]

(1) An operation error will occur and the error flag (SM0) will be turned ON in the following cases.

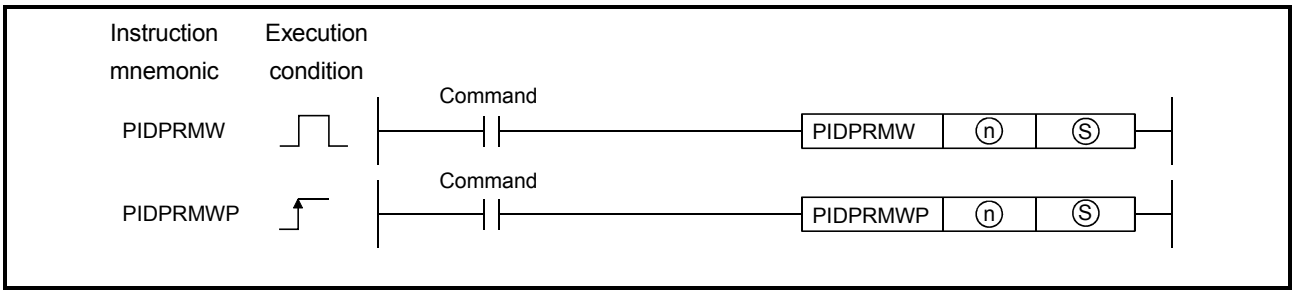
• When the loop number designated by ⓝ does not exist. (Error code: 4100)
• When n is outside the range 1 to 8. (Basic model QCPU) (Error code: 4100)
• When ⓝ is any other than 1 to 32. (High Performance model QCPU, Redundant CPU, Universal model QCPU, LCPU, and QnACPU) (Error code: 4100)
• When the 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)

Applicable CPU	QCPU					LCPU	QnA	Q4AR
	PLC CPU			Process CPU	Redundant CPU			
	Basic	High Performance	Universal					
	△*1	○	○	×	○	○	○	○

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

9.1.5 Parameter change at designated loop

Set Data	Usable Devices							
	Internal Devices (System, User)		File Register	MELSECNET/10 (H) Direct J□\□		Special Function Module U□\G□	Index Register Zn	Constant K, H	Other
	Bit	Word		Bit	Word				
ⓝ	○	○		○					—
Ⓢ	—	○		—					—



[SET DATA]

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

[FUNCTIONS]

(1) Changes the operation parameter for the loop number designated by ⓝ to the PID control data stored in the devices starting with the device number designated by Ⓢ.

(2) The configuration of the data for PID control which starts from the device number designated by Ⓢ is shown below.For details on PID control data, see Section 5.1.

Ⓢ +0	Selection of operation expression
Ⓢ +1	Sampling cycle (T_S)
Ⓢ +2	Proportional constant (K_P)
Ⓢ +3	Integral constant (T_I)
Ⓢ +4	Derivative constant (T_D)
Ⓢ +5	Filter coefficient (α)
Ⓢ +6	MV lower limit (MVLL)
Ⓢ +7	MV upper limit (MVHL)
Ⓢ +8	MV change rate limit (△MVL)
Ⓢ +9	PV change rate limit (△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.

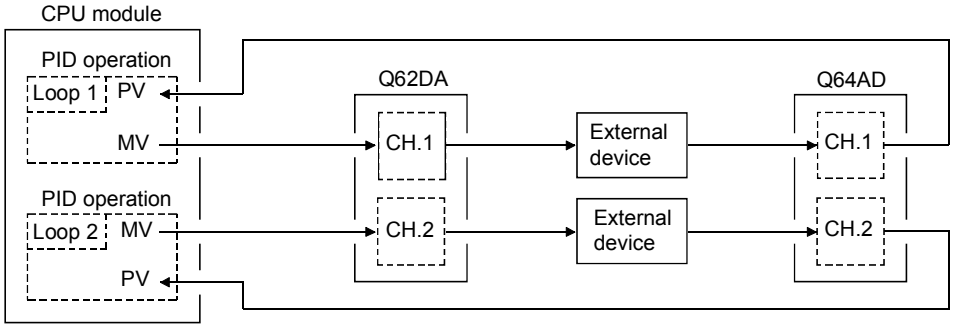
- When the loop number designated by ⓝ does not exist. (Error code: 4100)
- When ⓝ is outside the range 1 to 8. (Basic model QCPU) (Error code: 4100)
- When ⓝ is any other than 1 to 32. (High Performance model QCPU, Redundant CPU, Universal model QCPU, LCPU, and QnACPU) (Error code: 4100)
- When the PID control data is outside the setting range. (Error code: 4100)
- When the device range assigned to the PID control data area by Ⓢ 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)

9.2 PID Control Program Examples (for QCPU and LCPU)

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 numbers .....X/Y80 to X/Y8F
Q62DA I/O numbers .....X/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.

(4) The PID control data is set in the following devices:*1
Common data..... D500 and D501
Loop 1 data ..... D502 to D511
Loop 2 data ..... D512 to D521

(5) The I/O data is set in the following devices:*2
Common data..... D600 to D609
Loop 1 data ..... D610 to D627
Loop 2 data ..... D628 to D645

(6) The following SV are set for loop 1 and loop 2 using a sequence program:
Loop 1..... 600
Loop 2..... 1000

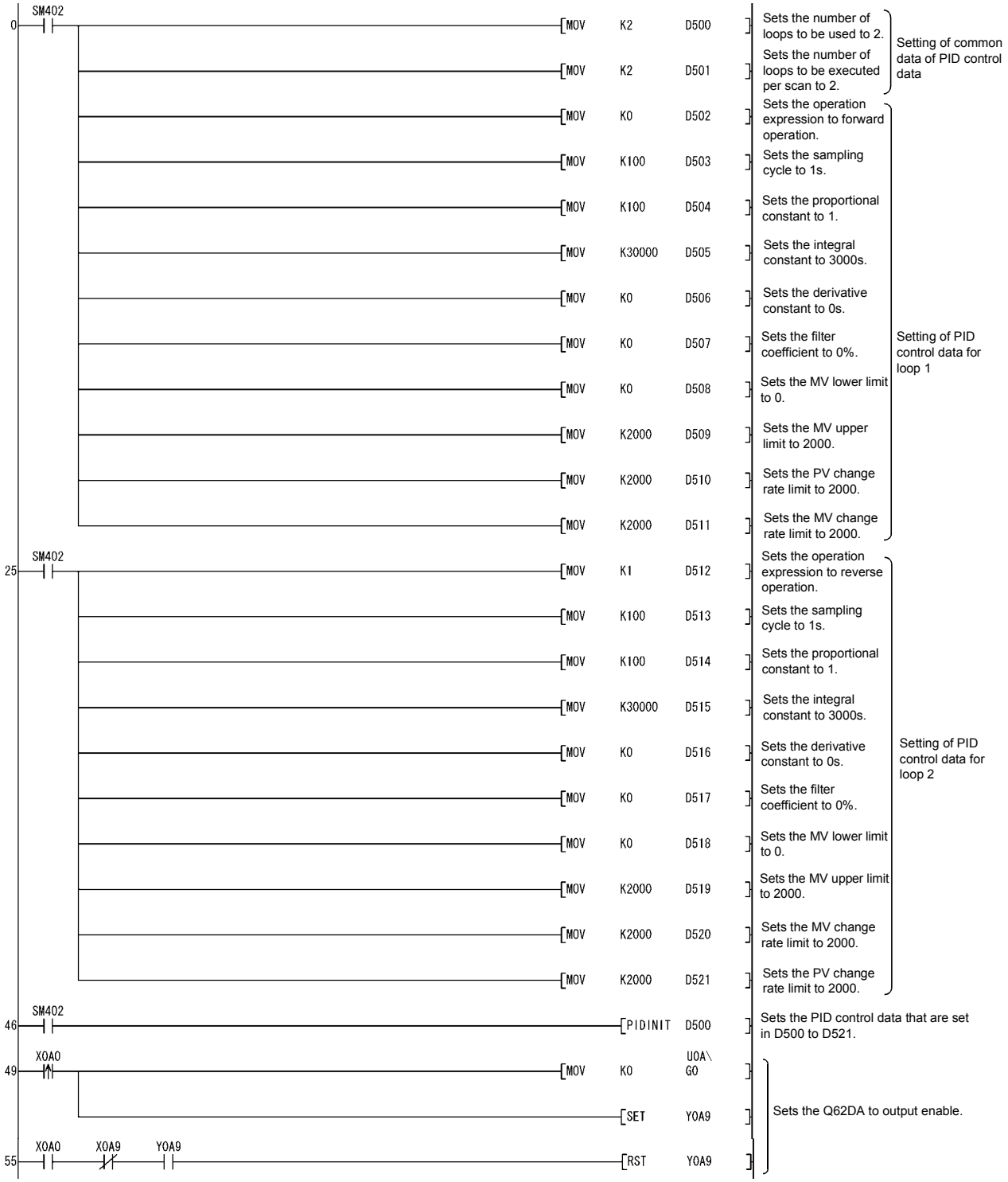
(7) The following devices are used for PID control start/stop commands.
PID control start command X0
PID control stop command X1

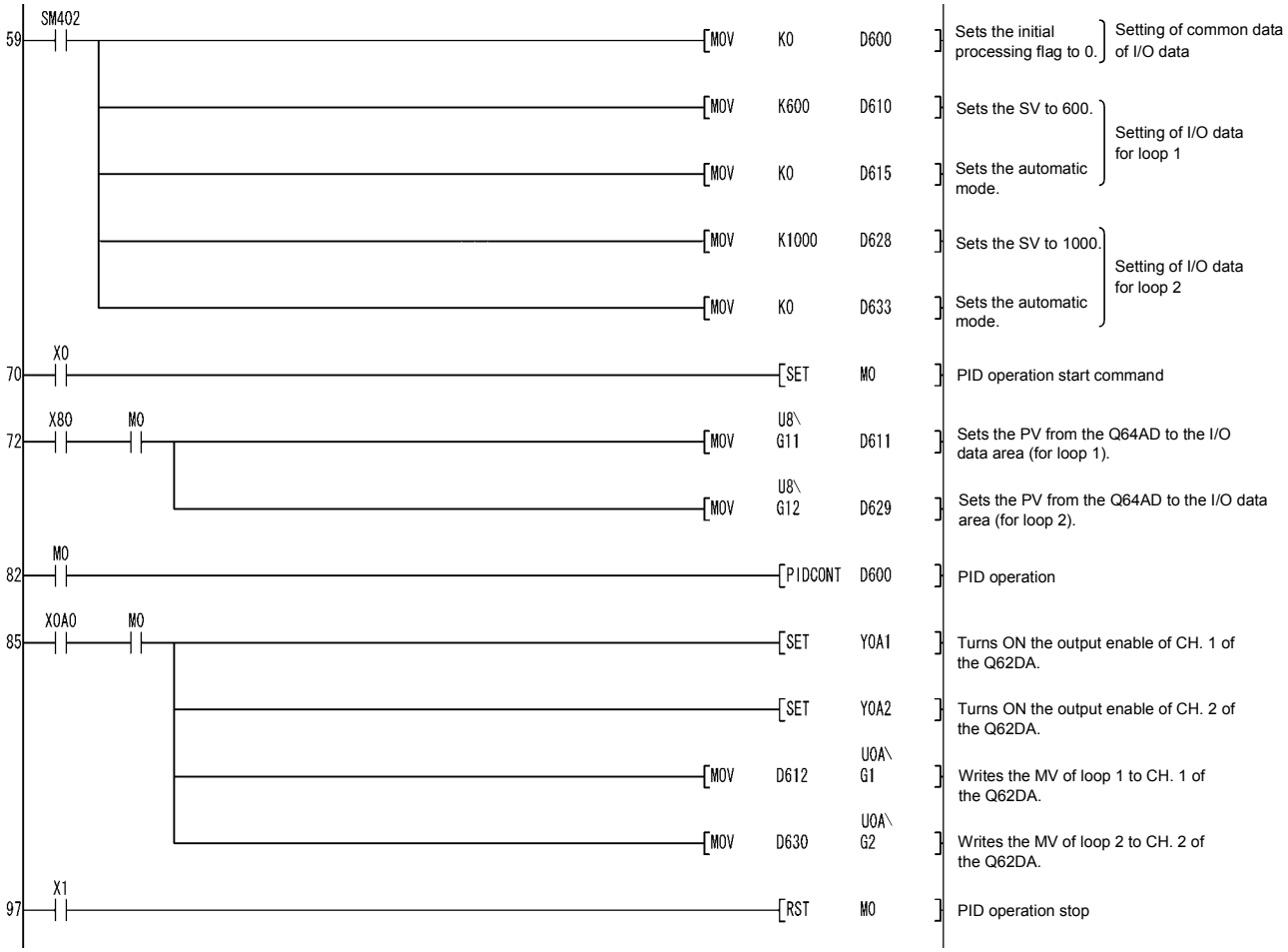
(8) The digital values of the Q64AD and Q62DA are set within the range 0 to 2000.

REMARK

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

[PROGRAM EXAMPLE]




59
SM402
MOV K0 D600
Sets the initial processing flag to 0.
Setting of common data of I/O data
MOV K600 D610
Sets the SV to 600.
MOV K0 D615
Sets the automatic mode.
Setting of I/O data for loop 1
MOV K1000 D628
Sets the SV to 1000.
MOV K0 D633
Sets the automatic mode.
Setting of I/O data for loop 2
70
X0
SET M0
PID operation start command
72
X80
M0
MOV U8\G11 D611
Sets the PV from the Q64AD to the I/O data area (for loop 1).
MOV U8\G12 D629
Sets the PV from the Q64AD to the I/O data area (for loop 2).
82
M0
PIDCONT D600
PID operation
85
X0A0
M0
SET Y0A1
Turns ON the output enable of CH. 1 of the Q62DA.
SET Y0A2
Turns ON the output enable of CH. 2 of the Q62DA.
MOV D612 U0A\G1
Writes the MV of loop 1 to CH. 1 of the Q62DA.
MOV D630 U0A\G2
Writes the MV of loop 2 to CH. 2 of the Q62DA.
97
X1
RST M0
PID operation stop

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 devices:
Common data..... D500 and D501
Loop 1 data ..... D502 to D511

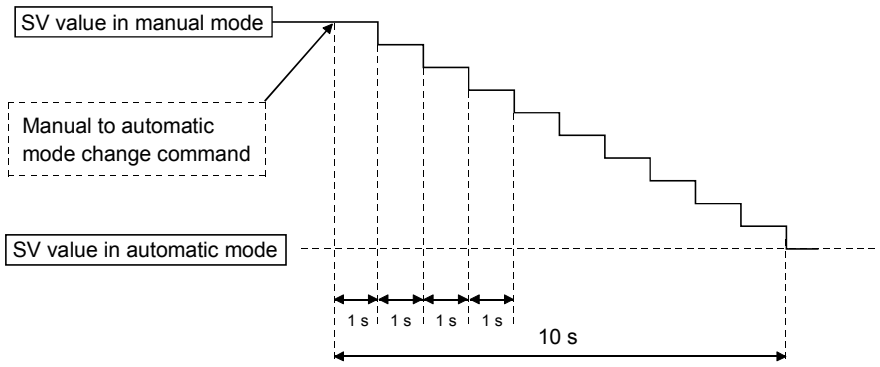
(5) The I/O data is set in the following devices:
Common data..... D600 to D609
Loop 1 data ..... D610 to D627

(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 stop PID control and the automatic/manual changeover command:
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 command 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.

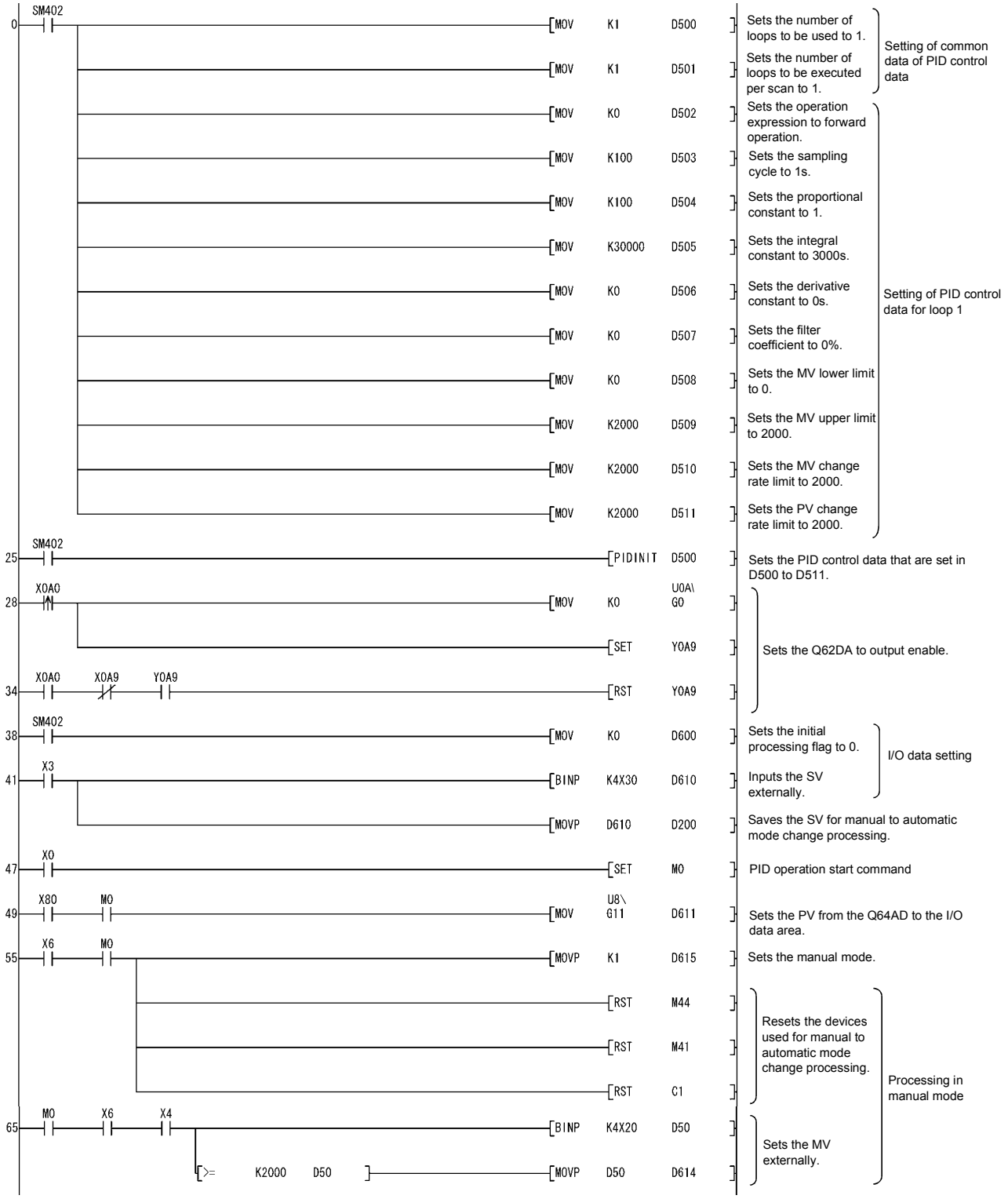


The SV is rewritten using the operation method illustrated below:

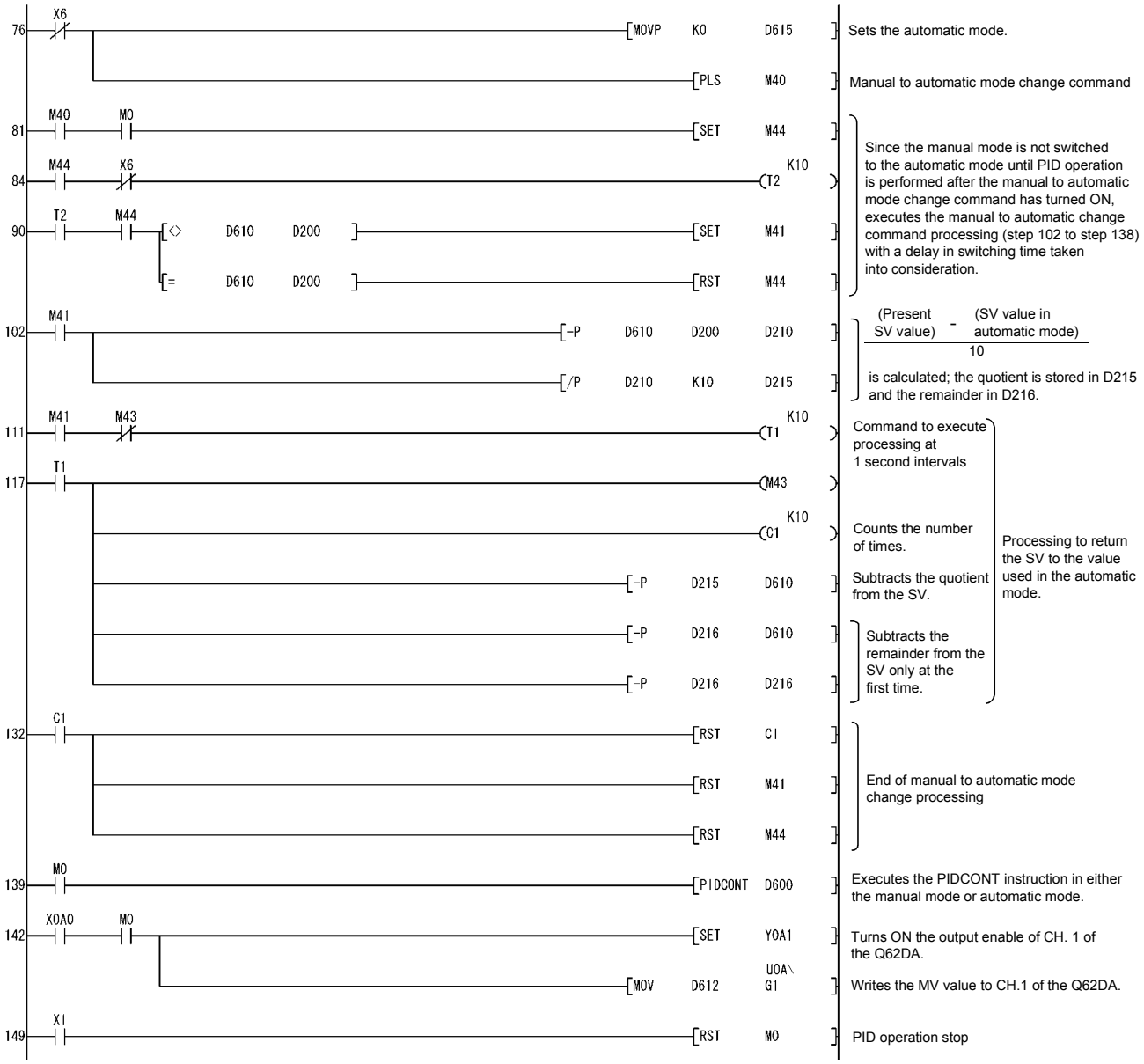
$$\frac{(\text{SV value in manual mode}) - (\text{SV value in automatice mode})}{10} = \boxed{\text{Incremental value}} \cdots\cdots \boxed{\text{Remainder}}$$

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

[PROGRAM EXAMPLE]



[PROGRAM EXAMPLE]

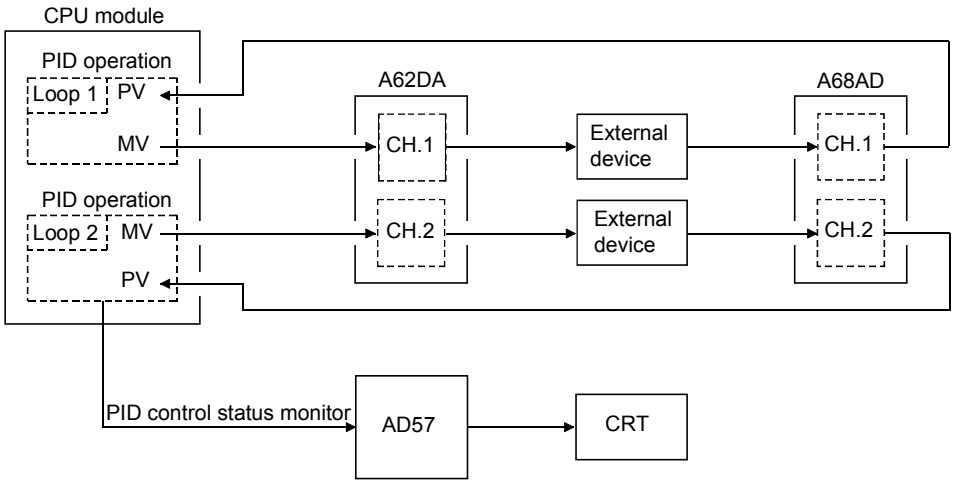


9.3 PID Control Program Examples (QnACPU only)

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.



A68AD I/O numbers.....X/Y80 to X/Y9F
A62DA I/O numbers.....X/YA0 to X/YBF
AD57 I/O numbers.....X/YC0 to X/YFF

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.

(4) The PID control data is set in the following devices:*1
Common data..... D500 and D501
Loop 1 data ..... D502 to D511
Loop 2 data ..... D512 to D521

(5) The I/O data is set in the following devices:*2
Common data..... D600 to D609
Loop 1 data ..... D610 to D627
Loop 2 data ..... D628 to D645

(6) The following SV are set for loop 1 and loop 2 using a sequence program:
Loop 1..... 600
Loop 2..... 1000

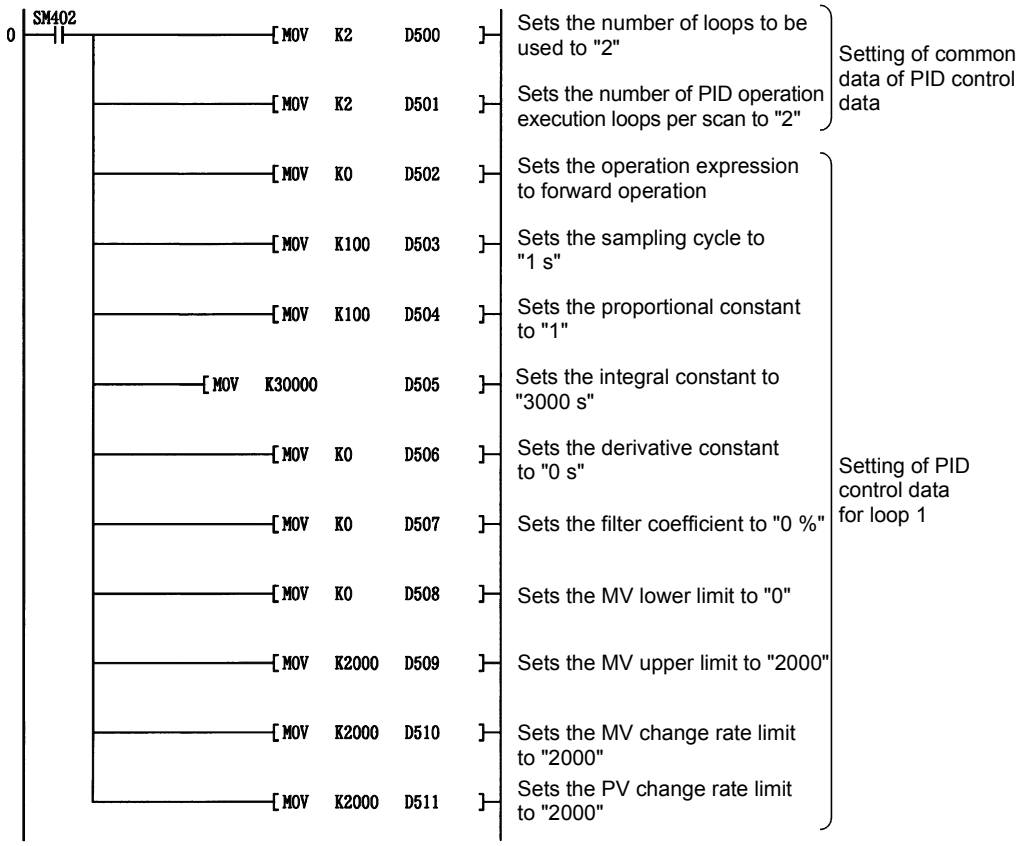
(7) The following devices are used for PID control start/stop commands and the monitoring command with AD57.
PID control start command X0
PID control stop command X1
Monitoring command with AD57 .... X2

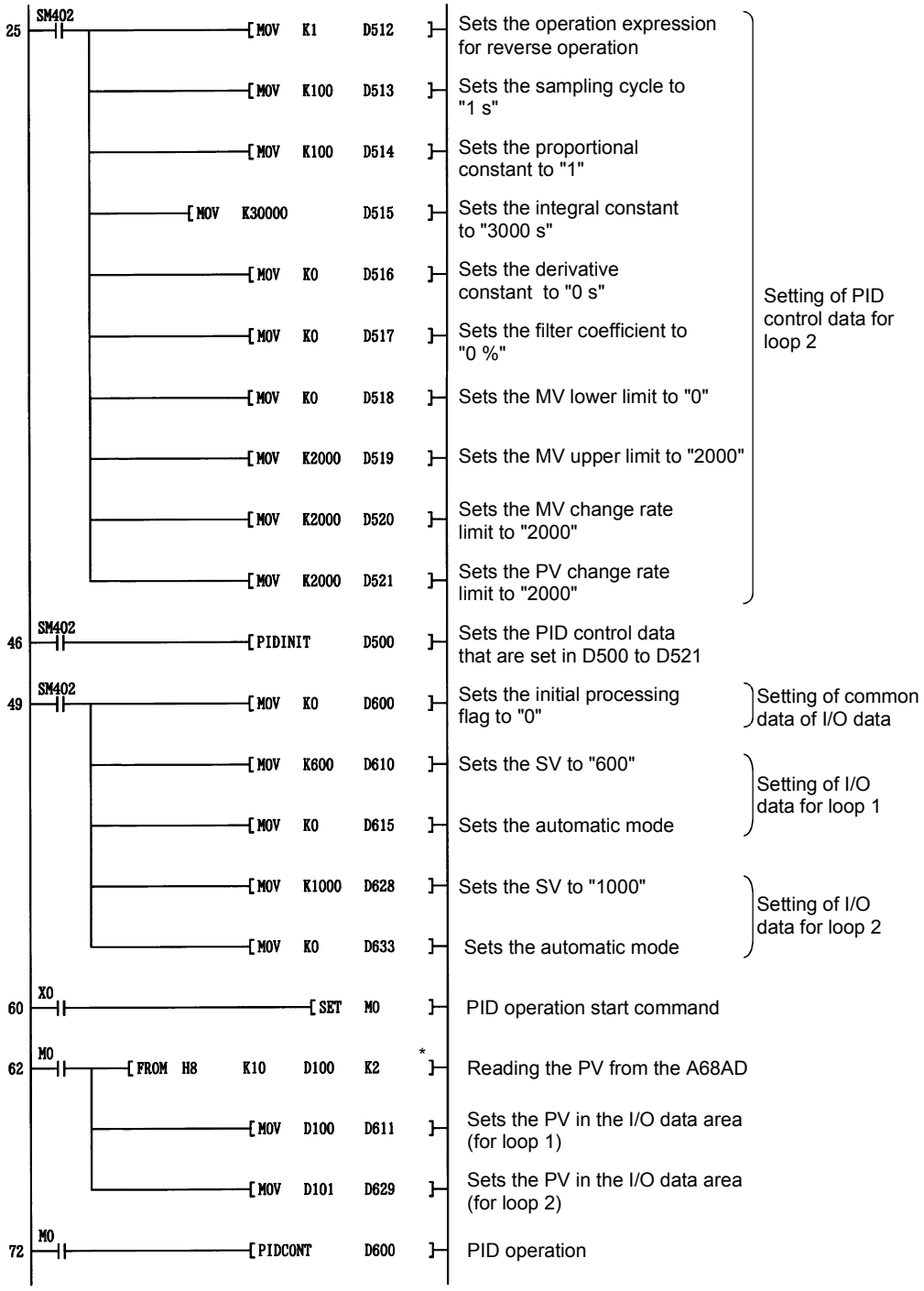
(8) The digital values of the A68AD and A62DA are set within the range 0 to 2000.

REMARK

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

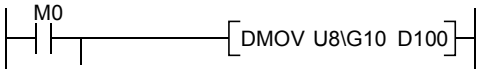
[PROGRAM EXAMPLE]

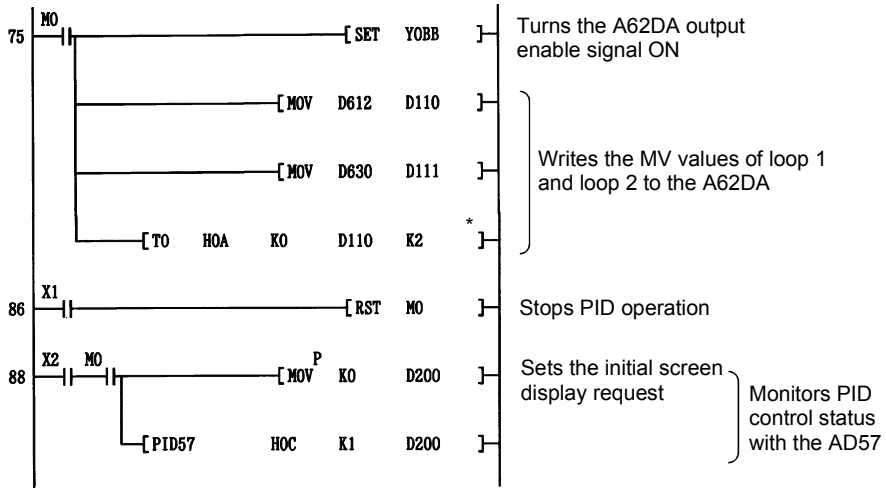




REMARK

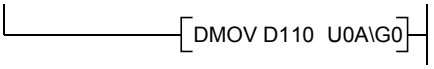
*: It is also possible to create a program by using special function module devices. In this case the format in the ladder is as follows:





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:



9.3.3 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

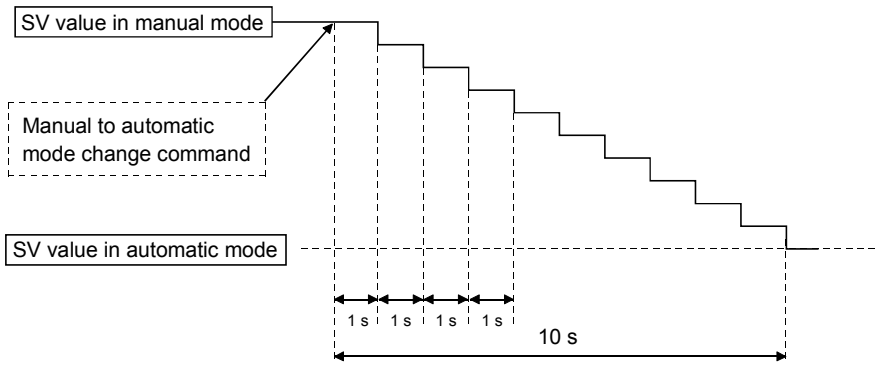
(5) The I/O data is set in the following devices:
Common data..... D600 to D609
Loop 1 data ..... D610 to D627

(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 and 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:

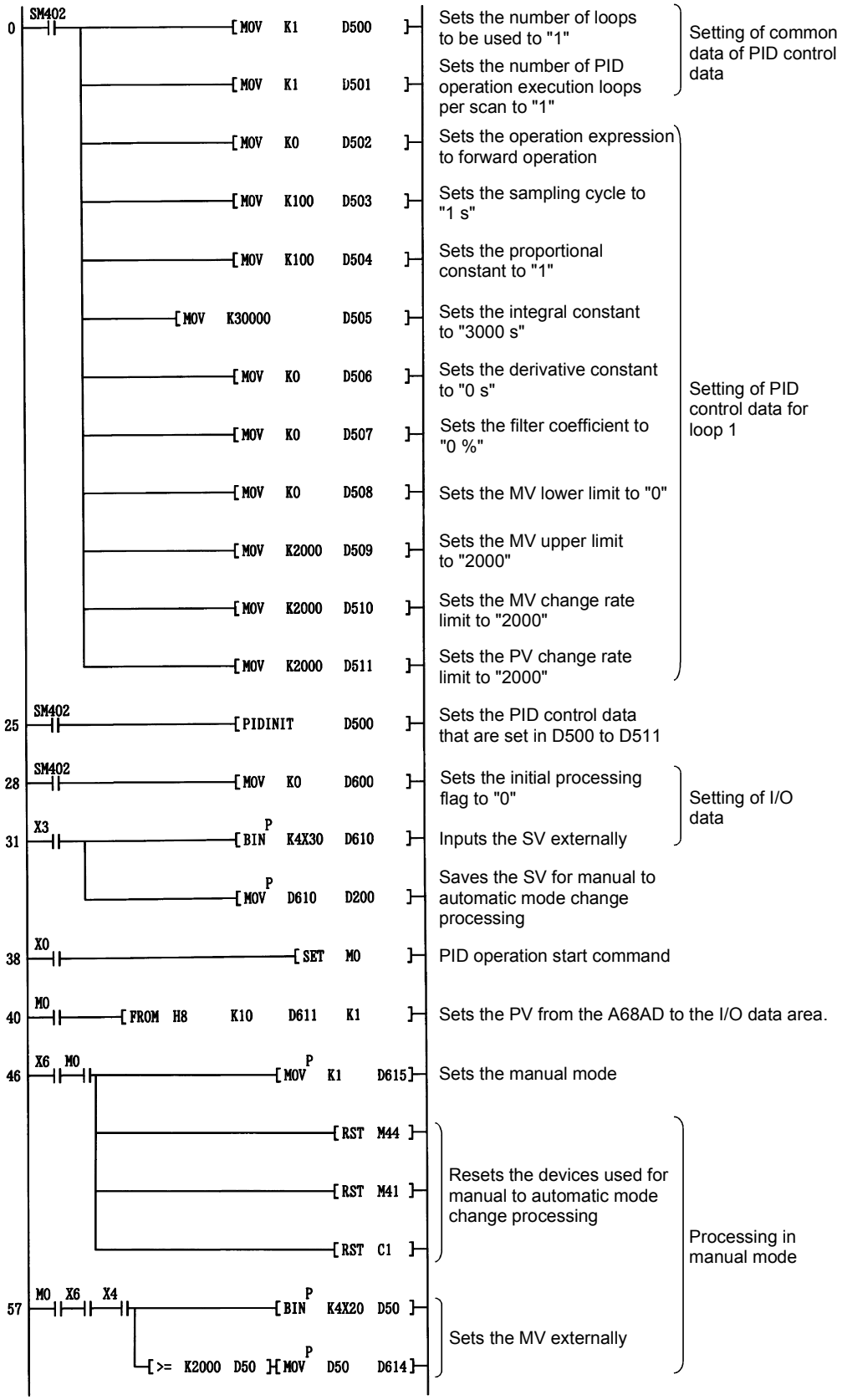


The SV is rewritten using the operation method illustrated below:

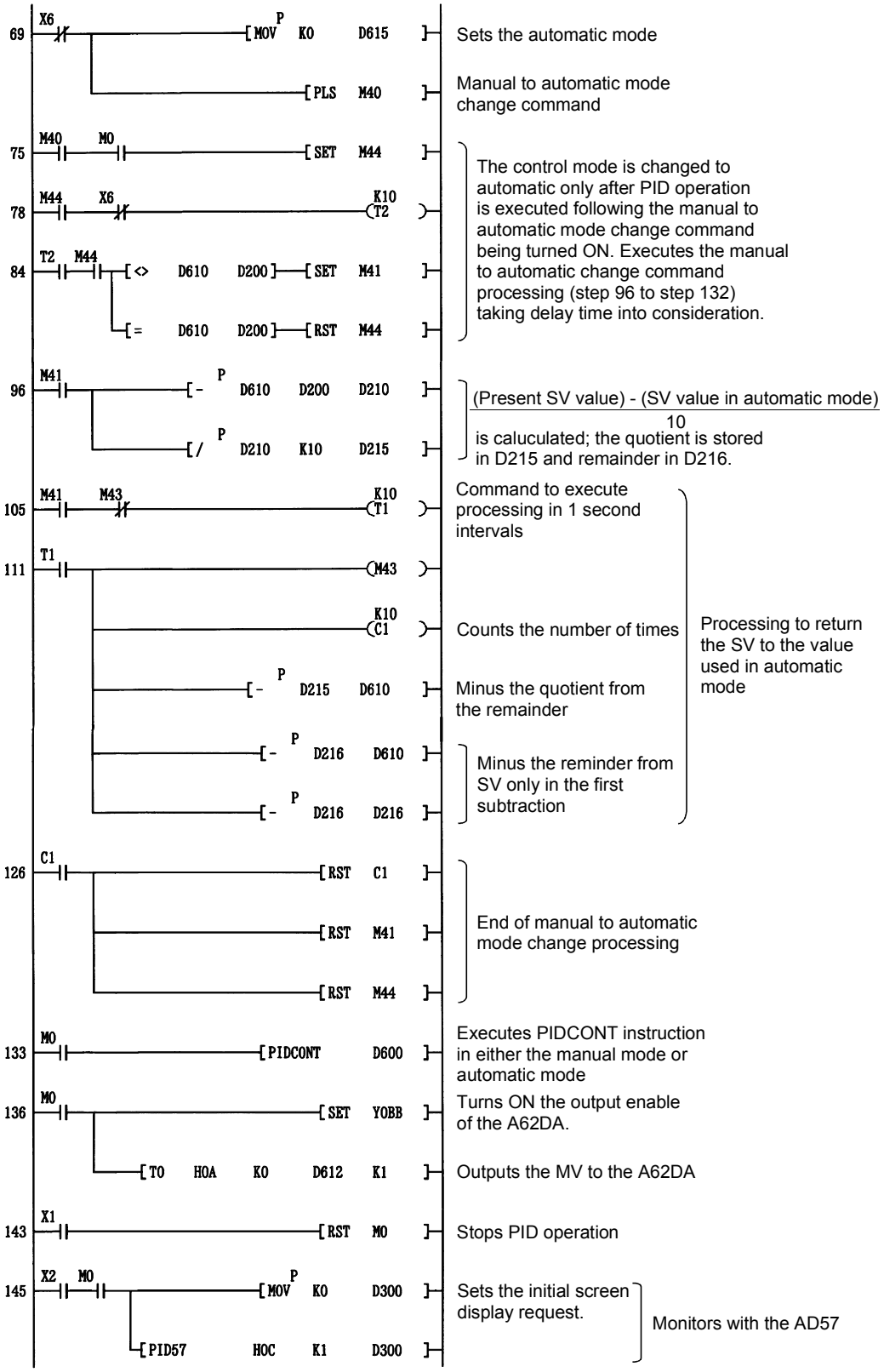
$$\frac{(\text{SV value in manual mode}) - (\text{SV value in automatice mode})}{10} = \boxed{\text{Incremental value}} \cdots\cdots \boxed{\text{Remainder}}$$

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

[PROGRAM EXAMPLE]



[PROGRAM EXAMPLE]



MEMO

APPENDIX

Appendix 1 Processing Time List

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

(a) Basic model QCPU, High Performance model QCPU, Redundant CPU, and Universal model QCPU

Instruction Name	Conditions		Processing Time (μs)					
			Q00JCPU	Q00CPU	Q01CPU	Q02CPU	QnHCPU	QnPRHCPU
S.PIDINIT	1 loop		115.0	97.0	88.5	64.5	28.0	28.0
	8 loops		250.0	210.0	190.0	—	—	—
	32 loops		—	—	—	410.0	180.0	180.0
S.PIDCONT	1 loop	First	395.0	335.0	300.0	215.0	92.0	92.0
		Others	350.0	300.0	270.0	190.0	81.5	81.5
	8 loop	First	2250.0	1850.0	1700.0	—	—	—
		Others	1950.0	1650.0	1500.0	—	—	—
	32 loops	First	—	—	—	4550.0	1950.0	1950.0
		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 Name	Conditions		Processing Time (μs)							
			Q00UJCPU Q00UCPU Q01UCPU		Q02UCPU		Q03UDCPU Q03UDECPU		Q04UDHCPU, Q06UDHCPU Q10UDHCPU, Q13UDHCPU Q20UDHCPU, Q26UDHCPU Q04UDEHCPU, Q06UDEHCPU Q10UDEHCPU, Q13UDEHCPU Q20UDEHCPU, Q26UDEHCPU	
			Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
S.PIDINIT	1 loop		17.5	39.9	14.2	48.2	14.2	22	11.4	18.8
	8 loops		—	—	—	—	—	—	—	—
	32 loops		295	376	230.1	298	231	261	164	183
S.PIDCONT	1 loop	First	123		114	131	50.5	52.5	46.8	47.5
		Others	90.4		77.4	111	39.5	47.2	35.8	42.1
	8 loop	First	—		—		—		—	
		Others	—		—		—		—	
	32 loops	First	1121		1082	1115	1036	1041	927	931
		Others	1046		852	883	820	842	728	744
S.PIDSTOP	1 loop		7.3	14.2	5.6	17.4	5.5	7.5	4.9	6.9
S.PIDRUN	1 loop		5.5	11.0	4.9	10.6	4.8	6.1	4.3	5.6
S.PIDPRMW	1 loop		18.7	62.4	13.3	33	13.0	16.7	10.7	14.5

(b) LCPU

Instruction Name	Conditions		Processing Time (μs)			
			L02CPU		L26CPU-BT	
			Min.	Max.	Min.	Max.
S.PIDINIT	1 loop		14.0	27.3	11.4	18.8
	8 loops		——	——	——	——
	32 loops		244.3	277	164	183
S.PIDCONT	1 loop	First	64.1	66.0	46.8	47.5
		Others	53.0	55.5	35.8	42.1
	8 loops	First	——		——	
		Others	——		——	
	32 loops	First	1064.2	1069.5	927	931
		Others	863.2	889.8	728	744
S.PIDSTOP	1 loop		6.9	10.6	4.9	6.9
S.PIDRUN	1 loop		6.2	15.2	4.3	5.6
S.PIDPRMW	1 loop		14.7	28.9	10.7	14.5

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

(a) Basic model QCPU, High Performance model QCPU, Redundant CPU, and Universal model QCPU

Instruction Name	Conditions		Processing Time (μs)								
			Q2ASCPU Q2ACPU(S1)	Q3ACPU	Q2ASHCPU(S1) Q4ACPU Q4ARCPU	Q00J CPU	Q00 CPU	Q01 CPU	Q02 CPU	QnH CPU	QnPRH CPU
PIDINIT	1 loop		61	46	23	66.0	56.0	50.5	26.0	11.2	11.2
	8 loops		—	—	—	170.0	145.0	130.0	—	—	—
	32 loops		407	306	153	—	—	—	174.0	74.9	74.9
PIDCONT	1 loop	First	211	159	80	325.0	275.0	245.0	86.6	37.3	37.3
		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	—	—	—
		Others	—	—	—	1700.0	1450.0	1300.0	—	—	—
	32 loops	First	5086	3824	1912	—	—	—	2102.5	904.9	904.9
		Others	4894	3680	1840	—	—	—	2036.9	876.7	876.7
PID57	1 loop	First	9629	7240	3620	—	—	—	—	—	—
		Others	606	456	228						
	8 loops	First	9669	7270	3635						
		Others	3719	2796	1398						
PIDSTOP PIDRUN	1 loop		11.2	8.4	4.2	22.0	18.5	17.0	4.5	1.9	1.9
PIDPRMW	1 loop		36	26	13	53.0	45.0	41.0	14.6	6.3	6.3

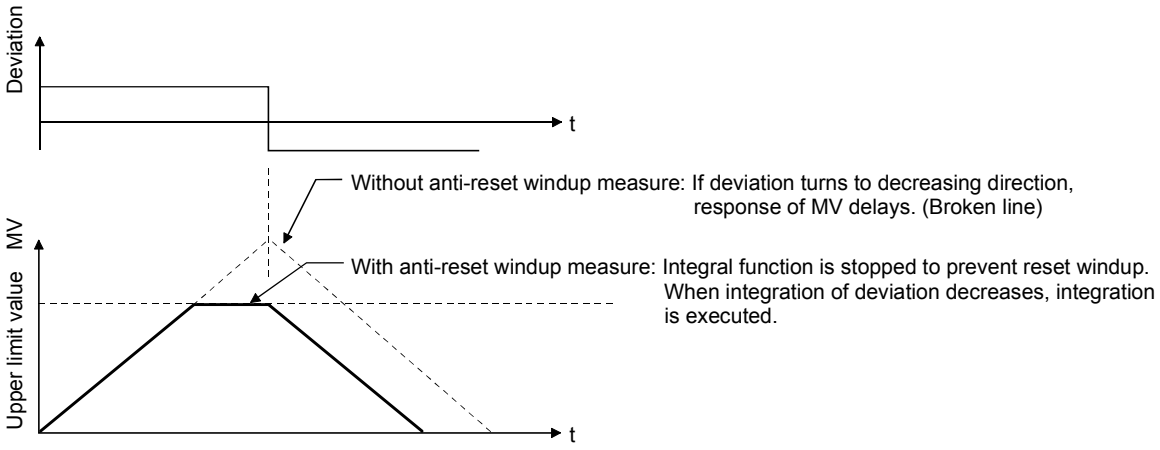
Instruction Name	Conditions		Processing Time (μs)							
			Q00UJCPU Q00UCPU Q01UCPU		Q02UCPU		Q03UDCPU Q03UDECPU		Q04UDHCPU, Q06UDHCPU Q10UDHCPU, Q13UDHCPU Q20UDHCPU, Q26UDHCPU Q04UDEHCPU, Q06UDEHCPU Q10UDEHCPU, Q13UDEHCPU Q20UDEHCPU, Q26UDEHCPU	
			Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
PIDINIT	1 loop		10.3	60.7	8.2	27.7	8.2	12.4	6.3	10.9
	8 loops		—		—		—		—	—
	32 loops		162	227	129	159	129	133	98.7	122.6
PIDCONT	1 loop	First	111		101	102	40.6	41.5	36.8	
		Others	95.5		51.5	74.4	33.6	38.5	35.8	
	8 loop	First	—		—		—		—	
		Others	—		—		—		—	
	32 loops	First	909		931	939	862	872	776	785
		Others	914		764	788	736	757	622	645
PIDSTOP	1 loop		3.3	33.4	1.8	6.7	1.8	3.4	1.3	2.9
PIDRUN	1 loop		1	4	1.7	6.7	1.7	3.2	1.5	2.7
PIDPRMW	1 loop		9.4	23	6.9	17.6	6.9	10.3	5.9	8.9

(b) LCPU

Instruction Name	Condition		Processing Time (μs)			
			L02CPU		L26CPU-BT	
			Min.	Max.	Min.	Max.
PIDINIT	1 loop		8.6	14.6	6.3	10.9
	8 loops		——	——	——	——
	32 loops		131.2	136.7	98.7	122.6
PIDCONT	1 loop	First	51.5	51.8	36.8	
		Others	50.0	50.4	35.8	
	8 loops	First	——		——	
		Others	——		——	
	32 loops	First	869.5	886.9	776	785
		Others	746.2	750.3	622	645
PIDSTOP	1 loop		2.2	4.0	1.3	2.9
PIDRUN	1 loop		2.2	4.2	1.5	2.7
PIDPRMW	1 loop		8.0	12.0	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 measure for the reset windup is taken in PID operation instructions (PIDCONT and S.PIDCONT) of the QCPU, LCPU, and QnACPU, integral operation need not be stopped.



WARRANTY

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company.

However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place.

Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

(1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.

(2) Even within the gratis warranty term, repairs shall be charged for in the following cases.

1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
2. Failure caused by unapproved modifications, etc., to the product by the user.
3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
5. Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
7. Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

2. Onerous repair term after discontinuation of production

(1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued. Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.

(2) Product supply (including repair parts) is not available after production is discontinued.

3. Overseas service

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

4. Exclusion of loss in opportunity and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation of damages caused by any cause found not to be the responsibility of Mitsubishi, loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products, special damages and secondary damages whether foreseeable or not , compensation for accidents, and compensation for damages to products other than Mitsubishi products, replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

MELSEC-Q/L/QnA
Programming Manual

PID Control Instructions

MODEL	QCPU(Q)-P(PI)-E
MODEL CODE	13JF59
SH(NA)-080040-M(1001)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.

Specifications subject to change without notice.