MELSEC A/Q Series

Programmable Logic Controllers

User's Manual

Q2ACPU(S1), Q3ACPU, Q4ACPU

Art. no.: 061740
01 07 1996
IB (NA)-66608

REVISIONS

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

Print Date	*Manual Number	Revision
Jul., 1996	IB (NA) 66608-A	First edition

SAFETY CAUTIONS

(You must read these cautions before using the product)

In connection with the use of this product, in addition to carefully reading both this manual and the related manuals indicated in this manual, it is also essential to pay due attention to safety and handle the product correctly.
The safety cautions given here apply to this product in isolation. For information on the safety of the PC system as a whole, refer to the CPU module User's Manual.
These **SAFETY CAUTIONS** are classified into two grades: "DANGER" and "CAUTION".

DANGER Safety caution given when incorrect handling could result in hazardous situations involving the possibility of death or serious injury.

CAUTION Safety caution given when incorrect handling could result in hazardous situations involving the possibility of moderate or light injury or damage to property.

Note that, depending on the circumstances, failing to follow a  **CAUTION** may also have very serious consequences.
Both of these classes of safety caution are very important and must be observed.
Store this manual carefully in a place where it is accessible for reference whenever necessary, and forward a copy of the manual to the end user.

[System Design Precautions]

DANGER

- Safety circuits should be installed external to the programmable controller to ensure that the system as a whole will continue to operate safely in the event of an external power supply malfunction or a programmable controller failure. Erroneous outputs and operation could result in an accident.
  1) The following circuitry should be installed outside the programmable controller:
     Interlock circuitry for the emergency stop circuit protective circuit, and for reciprocal operations such as forward/reverse, etc., and interlock circuitry for upper/lower positioning limits, etc., to prevent machine damage.
  2) When the programmable controller detects an abnormal condition, processing is stopped and all outputs are switched OFF. This happens in the following cases:
     - When the power supply module's over-current or over-voltage protection device is activated.
     - When an error (watchdog timer error, etc.) is detected at the PC CPU by the self-diagnosis function.

     Some errors, such as input/output control errors, cannot be detected by the PC CPU, and there may be cases when all outputs are turned ON when such errors occur. In order to ensure that the machine operates safely in such cases, a failsafe circuit or mechanism should be provided outside the programmable controller. Refer to the CPU module user's manual for an example of such a failsafe circuit.
  3) Outputs may become stuck at ON or OFF due to an output module relay or transistor failure. An external circuit should therefore be provided to monitor output signals whose incorrect operation could cause serious accidents.

- A circuit should be installed which permits the external power supply to be switched ON only after the programmable controller power has been switched ON. Accidents caused by erroneous outputs and motion could result if the external power supply is switched ON first.

- When a data link communication error occurs, the status shown below will be established at the faulty station. In order to ensure that the system operates safely at such times, an interlock circuit should be provided in the sequence program (using the communication status information).
  Erroneous outputs and operation could result in an accident.
  1) The data link data which existed prior to the error will be held.
  2) All outputs will be switched OFF at MELSECNET (II, /B, /10) remote I/O stations.
  3) At the MELSECNET/MINI-S3 remote I/O stations, all outputs will be switched OFF or output statuses will be held, depending on the E.C. mode setting.

  For details on procedures for checking faulty stations, and for operation statuses when such errors occur, refer to the appropriate data link manual.

[System Design Precautions]

⚠ CAUTION

- Do not bundle control lines or communication wires together with main circuit or power lines, or lay them close to these lines.
  As a guide, separate the lines by a distance of at least 100 mm, otherwise malfunctions may occur due to noise.
  When file register R that are outside the range are read, e.g. by a MOV instruction, the file register data will become $FFFF_H$ and use of this data will cause malfunctions. Take care not to use file registers that are outside the range when designing programs.
  For details on instructions, refer to the Programming Manual.

[Cautions on Mounting]

⚠ CAUTION

- Use the PC in an environment that conforms to the general specifications in the manual.
  Using the PC in environments outside the ranges stated in the general specifications will cause electric shock, fire, malfunction, or damage to/deterioration of the product.
- Make sure that the module fixing projection on the base of the module is properly engaged in the module fixing hole in the base unit before mounting the module.
  Failure to mount the module properly will result in malfunction or failure, or in the module falling.
- Extension cables should be securely connected to base unit and module connectors. Check for loose connection after installation.
  A poor connection could result in contact problems and erroneous inputs/outputs.
- Plug the memory card firmly into the memory card mounting connector. Check for loose connection after installation.
  A poor connection could result in erroneous operation.

[Cautions on Wiring]

DANGER

- Switch off the external power supply before staring installation and wiring work.
  Failure to do so could result in electrical shocks and equipment damage.
- After installation and wiring is completed, be sure to attach the terminal cover before switching the power ON and starting operation.
  Failure to do so could result in electrical shocks.

CAUTION

- Be sure to ground the FG and LG terminals, carrying out at least class 3 grounding work with a ground exclusive to the PC.
  Otherwise there will be a danger of electric shock and malfunctions.
- Carry out wiring to the PC correctly, checking the rated voltage and terminal arrangement of the product.
  Using a power supply that does not conform to the rated voltage, or carrying out wiring incorrectly, will cause fire or failure.
- Outputs from multiple power supply modules should not be connected in parallel. Failure to do so could cause the power supply module to overheat, resulting in a fire or module failure.
- Tighten the terminal screws to the stipulated torque.
  Loose screws will cause short circuits, fire, or malfunctions.
- Make sure that no foreign matter such as chips or wiring offcuts gets inside the module.
  It will cause fire, failure or malfunction.
- Connectors for external connections should be crimped, pressure welded, or soldered in the correct manner using the correct tools.
  For details regarding crimping and pressure welding tools, refer to the input/output module user's manual.
  A poor connection could cause shorts, fire, and erroneous operation.

[Cautions on Startup and Maintenance]

DANGER

- Do not touch terminals while the power is ON.
  This will cause malfunctions.
- Make sure that the battery is connected properly. Do not attempt to charge or disassemble the battery, do not heat the battery or place it in a flame, and do not short or solder the battery.
  Incorrect handling of the battery can cause battery heat generation and ruptures which could result in fire or injury.
- Switch the power off before cleaning or re-tightening terminal screws.
  Carrying out this work while the power is ON will cause failure or malfunction of the module.

CAUTION

- In order to ensure safe operation, read the manual carefully to acquaint yourself with procedures for program changes, forced outputs, RUN, STOP, and PAUSE operations, etc., while operation is in progress.
  Incorrect operation could result in machine failure and injury.
- Do not disassemble or modify any module.
  This will cause failure, malfunction, injuries, or fire.
- Switch the power OFF before mounting or removing the module.
  Mounting or removing it with the power ON can cause failure or malfunction of the module.
- When replacing fuses, be sure to use the prescribed fuse. A fuse of the wrong capacity could cause a fire.

[Cautions on Disposal]

CAUTION

- Dispose of this product as industrial waste.

INTRODUCTION

Thank you for choosing the Mitsubishi MELSEC-QnA Series of General Purpose Programmable 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

About Manuals:

Other manuals related to QnACPU operation (shown below) are also available if necessary.

Related Manuals

Manual Name	Manual No.
QnACPU Guidebook This manual is designed for first-time users of the QnACPU. It explains the procedures for all operations from program creation, to program writing to the CPU, and program debugging. It also explains how to use the QnACPU special features.	IB-66606
QnACPU Programming Manual (Fundamentals) This manual explains the programming procedures required for program creation. It also explains the device names, parameters, and program types. (optional)	IB-66614
QnACPU Programming Manual (Common Instructions) This manual explains how to use the sequence instructions, basic instructions, and application instructions. (optional)	IB-66615
QnACPU Programming Manual (Special Function Module) This manual explains the dedicated instructions used with special function modules at the Q2ACPU(S1), Q3ACPU, and Q4ACPU. (optional)	IB-66616
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. (Purchased separately)	IB-66617
QnACPU Programming Manual (PID Control Instructions) This manual explains the dedicated instructions used to execute PID control at the Q2ACPU(S1), Q3ACPU, and Q4ACPU. (optional)	IB-66618
QnACPU Programming module (SFC) This manual explains the SW0IVD-SAP3 system configuration, performance specifications, functions, programming, debugging, and error codes. (optional)	IB-66619
Building Block Type I/O Module User's Manual Describes the specifications of building block type I/O modules. (Purchased separately)	IB-66140
MELSECNET/10 Network System (for QnA) Reference Manual Describes the gnenral concept, specifications, and part names and settings, for MELSECNET/10. (Purchased separately)	IB-66620
MELSECNET, MELSECNET/B Data LInk System Reference Manual Describes the general concept, specifications, and part names and settings, for MELSECNET(II), MELSECNET/B. (Purchased separately)	IB-66350
Type SW0IVD-GPPQ GPP Function Operating Manual (OFFLINE) Describes the how to create programs and prin out data when using SW0IVD-GPPQ, and the offline functions fo SW0IVD-GPPQ such as file maintenance. (Supplied with the product)	IB-66623
Type SW0IVD-GPPQ GPP Function Operating Manual (ONLINE) Describes the online functions of SW0IVD-GPPQ, including the methods for monitoring and debugging. (Supplied with the product)	IB-66624
Type SW0IVD-GPPQ GPP FunctionOperating Manual (SFC) Describes the system configuration, performance specifications, functions, system startup procedure, SFC program editing method, monitoring method, printout method, and error messages, for MELSAP-3. (Supplied with the product)	IB-66625

1. ABOUT THIS MANUAL

1.1 About this Manual

This manual serves to explain the specifications and functions of the Q2ACPU(S1), Q3ACPU, and Q4ACPU (hereafter collectively called "QnACPU"), the specifications of other modules, and the maintenance required for smooth system operation, to users of MELSEC-QnA series programmable controllers.

It is divided into the following three main parts:

(1)	Sections 2 and 3	These sections give the general description and system configuration for the QnACPU. Read them to learn the features of QnACPU, and the modules that can be used and points to note when configuring a system.
(2)	Sections 4 to 15	These sections give the specifications and functions of QnACPU. They describe each QnACPU function to enable you to use the QnACPU effectively.
(3)	Sections 16 to 18	These sections describe the specifications and handling of units and modules other than the CPU module (power supply module, base units, etc.). Read them to learn how to handle the power supply module, base units, memory cards, etc.
(4)	Sections 19 to 21	These sections describe all aspects of maintenance, from installing the QnACPU to daily inspections and troubleshooting. Read them to learn how to install the QnACPU so as to ensure smooth operation, and how to carry out daily inspections and corrective action in the event of trouble.

REMARK

This manual does not cover MELSECNET(II) data link systems, MELSECNET/B data link systems, MELSECNET/10 networks, or the SFC function.
For details on functions relating to these, refer to the following manuals.

- MELSECNET(II), MELSECNET/B Data Link
  MELSECNET, MELSECNET/B Data Link System Reference Manual
- MELSECNET/10 Network
  MELSECNET/10 Network System Manual for QnA
- SFC Function
  QnACPU Programming Manual (SFC)

POINT

In this manual, the RAM memory built into the QnACPU is referred to as the "internal memory".
Read the "Internal RAM" displayed on SW□IVD-GPPQ screens as "internal memory".

1.2 Abbreviations and Generic Terms Used in this Manual

The following abbreviations and generic terms are used in this manual.

(1) QnACPU	Abbreviation for Q2ACPU, Q2ACPU-S1, Q3ACPU, and Q4ACPU CPU modules.
(2) GPPQ, GPP function software	Abbreviations for the SW0IVD-GPPQ type GPP function software package.
(3) IBM PC/AT	IBM's PC/AT or completely compatible computers (*1).
(4) Peripheral device capable of GPP functions	Generic term for a peripheral device capable of running the GPP function software, for example a IBM PC/AT.
(5) Q6PU	Abbreviation for Q6PU programming unit.
(6) Peripheral device	Generic term for a device that is connected to a QnACPU and can be used to operate it, for example a IBM PC/AT or Q6PU.
(7) Internal memory	A RAM incorporated in the QnACPU that stores sequence programs and other data.
(8) Memory card	Abbreviation for Q1MEM-□□□ type memory card.
(9) ACPU	Generic term for a MELSEC-A series programmable controller.

*1: IBM is registered trademark of International Business Machines Corporation.

2. GENERAL DESCRIPTION

2.1 Features

The QnACPU has the following features.

(1) Large memory capacity

(a) The Q4ACPU has a program capacity of 124 ksteps, which means that 124 ksteps can be used for a single program (Q2ACPU: 28 ksteps, Q2ACPU-S1: 60 ksteps, Q3ACPU: 92 ksteps).

(b) The device memory capacity is 32 kwords and the user can change the number of points as required.

For example, the default number of points for internal relays (M) is 8 kpoints, but this can be expanded to up to 32 kpoints.

(c) Two memory cards of a maximum of 2 Mbytes can be installed.

Memory cards are used to store programs, comments, statements, and file registers.
(Programs can be stored in the CPU itself, so a memory card is not essential to run a QnACPU.)

(2) High-speed processing

(a) Higher operation processing speeds have been achieved for basic instructions and application instructions.

	A4UCPU		Q4ACPU
Basic instructions	0.15 μs	→	0.075 μs
Application instructions	0.90 μs	→	0.225 μs

(b) The access time for expansion data memory (file registers: R) has been made the same as the internal devices of the QnACPU (data registers: D, and link registers: W).

(c) Reading/writing of the buffer memories of special function modules dedicated to QnA (serial communication modules) are 6 times faster than for AnUCPU.

(The processing speed for the existing special function modules for ACPU use is about the same as that when using AnUCPU.)

(d) A high-speed access base unit (A38HB) is available to speed up the processing time for accessing special function modules such as network modules and serial communication modules that handle large quantities of data.

Simply by mounting the special function module on the high-speed access base unit, the access processing speed when the special function module is accessed from the QnACPU is increased.

(3) It has been made possible to select the type of program execution that is appropriate for the control.
The most appropriate program execution type can be selected from the following four types.

(a) Initial execution type

This program type is executed once only when the QnACPU is set to RUN.

(b) Scan execution type

This program type is run continually while the QnACPU is in the RUN state.
It is the same as a conventional program that is run from step 0 to the END instruction. It is possible to create subroutine programs and interrupt programs for this type of program.

(c) Low-speed execution type

This program type is executed in the surplus time when using constant scan time (processing in which the program execution time is set in advance and the scan time kept constant).

(d) Standby type

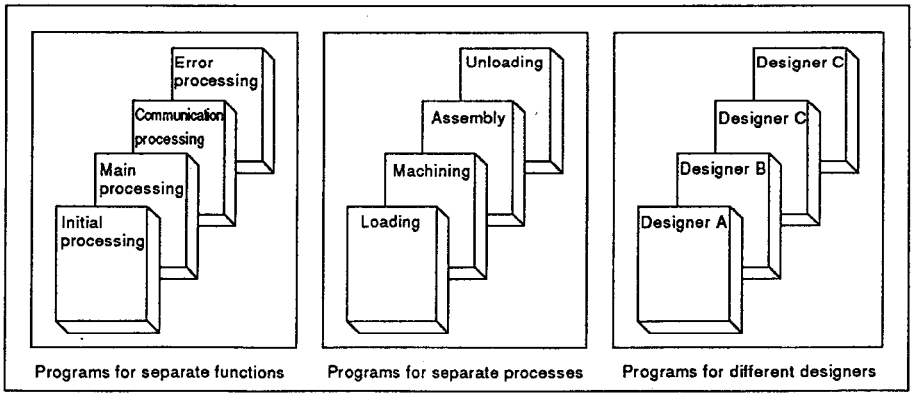
This type of program consists entirely of a subroutine program or interrupt program.

(4) The SFC language MELSAP3 has been made a standard feature.
With comprehensive step attributes and SFC control instructions, MELSAP3 makes SFC programming even easier.

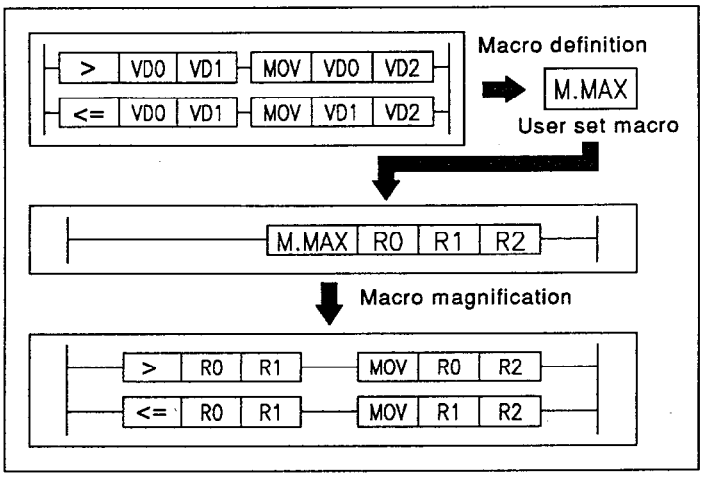
(5) A software development environment that improves program productivity has been realized.

(a) In order to enable the design of structured programs, a file format has been adopted for programs.

What would conventionally have been a single continuous program can now be managed in a structured way as a number of files.
This makes it possible for design work to be shared by several designers, and allows management of programs in accordance with functions, processes, or designers, etc.



(b) The user can standardize and simplify programs by creating and using macro instructions corresponding to functions.



(c) Devices can be used without restrictions.

1) Word device bit operations are possible.
2) Differential contacts can be used.
3) Buffer memories of special function modules can be accessed directly from a program as devices.
4) The link data of network modules can be accessed directly from a program as devices.

(d) Ease of operation for GPPQ program editing has been improved.

1) Up to four programs, data, etc., can be edited simultaneously. Programs and data can be cut and pasted between edited objects.
2) Ladder editing is possible while the ladder is displayed with comments.
3) Familiar operations can be performed with pull-down menus and dialog boxes.

(e) The debugging function at startup has been perfected.

1) Ladder modification while performing monitoring is possible.
2) Coil ON/OFF causes can be searched for.
3) The timing for monitoring can be set using a step number or device status, allowing debugging to be conducted under the optimum conditions.
4) Devices for which index qualifications have been set can be monitored.

(f) The GPPQ document creation function has been strengthened.

1) Since comments can now comprise 32 characters, they can be more detailed than before.

2) Comments can now be set for all devices.

3) The statements and notes appended to programs can now be managed as an integral part of the program, which makes program modifications and utilization easier.

4) Printout data can be saved in a file, reducing printout time.

(g) A powerful array of support software packages is available for program creation.

1) Data conversion package
Comment data, device data, etc., created with spreadsheet software and text editors available on the market, can be converted to files for GPPQ use.
Conversely, files created for GPPQ use can be converted to data for spreadsheets or text editors.

2) Macro/library package
The basic programs for accessing special function modules, and standard programs for error detection, alarm processing, etc., have been brought together as a package of macro and library data.

3) Ladder sequence linking package
This package is used to link multiple programs to make a single program.
It features an automatic allocation function that ensures that devices from each program are not duplicated in the created program.

4) CAD interface program
This package is used to handle sequence ladders, instruction lists, comment data and SFC diagrams as CAD data and communicate this data to CAD systems.

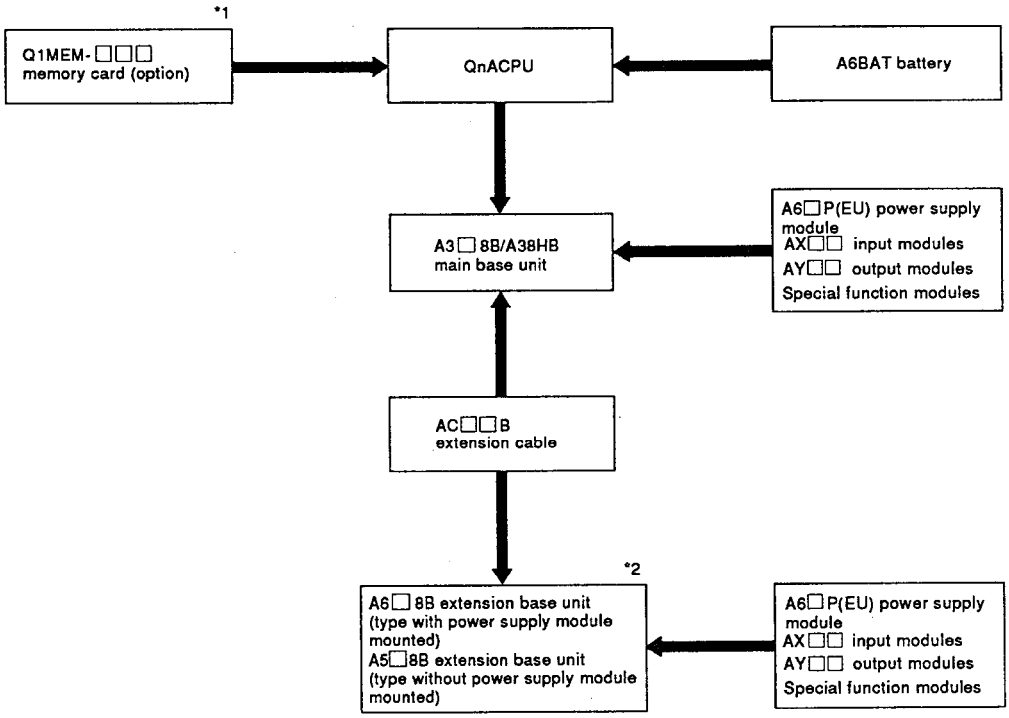
3. SYSTEM CONFIGURATION

This section describes the system configurations that can be used for a system centered on a QnACPU, cautions on configuring the system, and the system equipment.

3.1 System Configuration

The equipment configuration and peripheral device configuration when a QnACPU is used in an independent system is described here.

3.1.1 Equipment configuration in an independent system

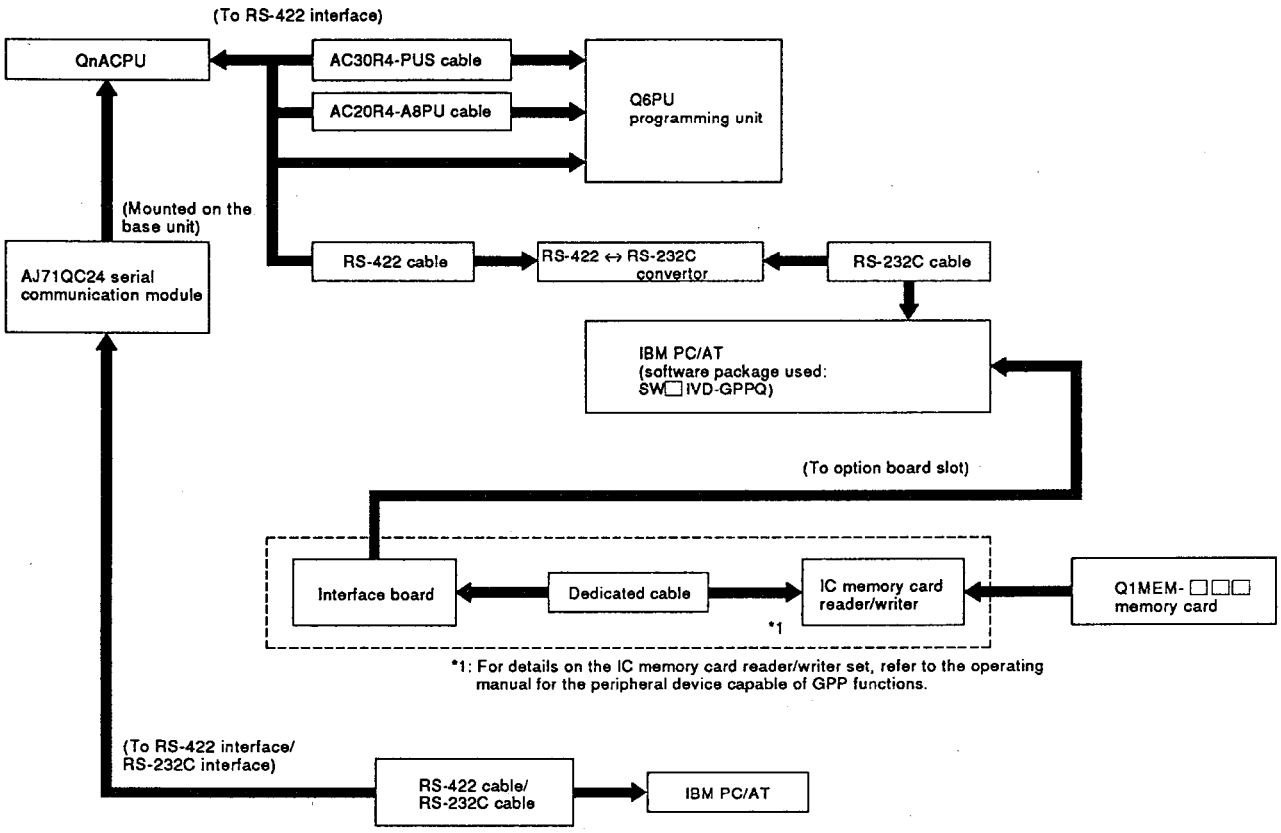


POINTS

*1 Up to two memory cards can be installed, if required.
The SRAM and E^2PROM memory cards are installed in the CPU module and can only be used for file reading. To write files, use a peripheral device (see Section 3.1.2).

*2 When using an A5□B extension base unit, pay particular attention to the power supply capacity of the main base unit. In the case of I/O modules with a high internal current consumption, you are recommended to mount the special function module on an A6□B extension base unit. (For details, see Sections 16.1.2 and 17.1.3.)

3.1.2 Configuration of peripheral devices for QnACPU

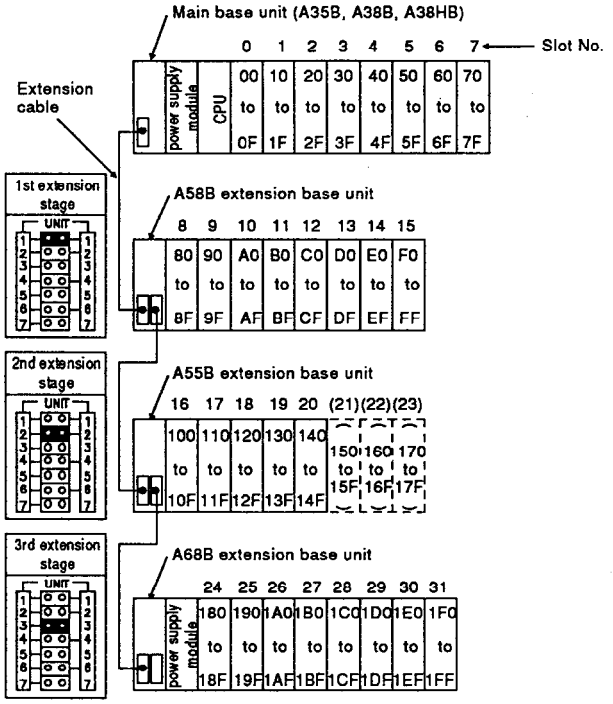


REMARKS

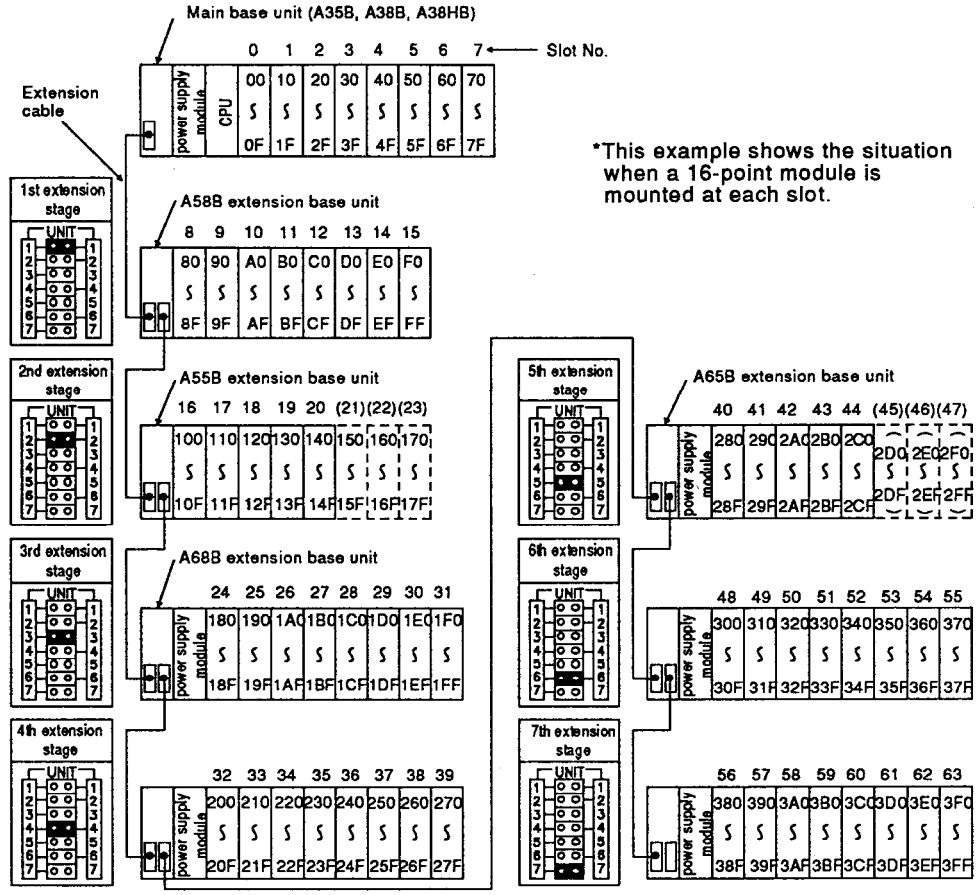
1. For details on the system configuration for each peripheral device, refer to the operating manual for the peripheral device in question.
2. It is only possible to connect a peripheral device for use with ACPU to a QnACPU when accessing an ACPU at another station via a MELSECNET data link. (However, in this case it is not possible to access the QnACPU.) In this case, set SW2 of system setting switch 2 on the CPU module ON.

3.2 Guide to System Configuration

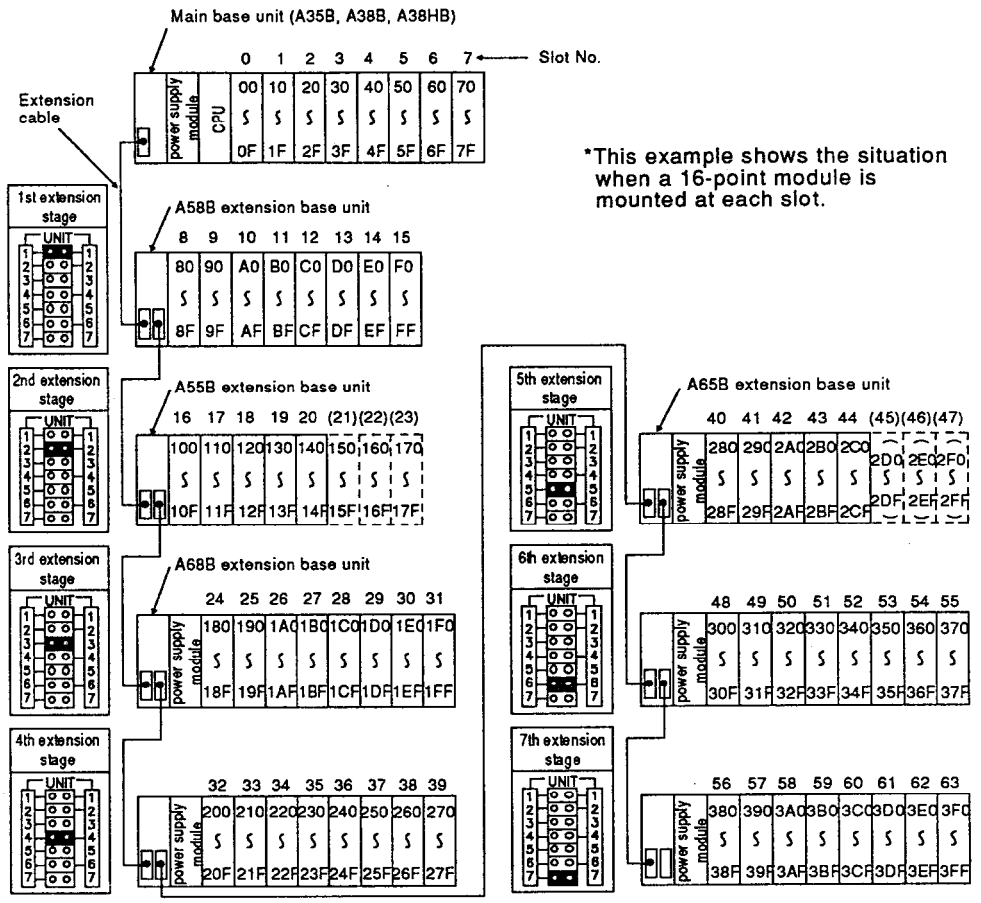
(a) Q2ACPU system

System configuration	 * This example shows the situation when a 16-point module is mounted at each slot.
Maximum number of extension stages	3rd extension stage
Maximum number of I/O modules	32 modules
Maximum number of I/O points	512 points
Main base/extension base types	A32B, A35B, A38B, A38HB/A62B, A65B, A68B, A52B, A55B, A58B
Extension cable types	AC06B, AC12B, AC30B
Restrictions	(1) It is not possible to connect an extension base unit to an A32B main base unit (there is no connector for extensions on the A32B). (2) When using an A52B, A55B, or A58B extension base unit, since 5 VDC is supplied from the power supply module on the main base unit, refer to Section 17.3 for details on use and to determine whether or not it is feasible to use these units. (3) The overall distance of the extension cable must not be longer than 6.6 m (21.65 ft).
Allocation of I/O numbers	(1) I/O numbers are allocated in accordance with the order of the numbers set for the extension base stages, regardless of the order in which the extension cables are connected. (2) I/O numbers are allocated on the assumption that all main base units and extension base units have 8 slots. Consequently, allocations of 16 points per slot are made for the parts of the system configuration drawing indicated with dotted lines. (3) Allocate 16 points to vacant slots. (4) Extension stage numbers do not need to be sequential, but any extension stage omitted from the system will occupy 8 (slots) x 16 I/O points. (5) With regard to items (2) to (4) above, the I/O allocations can be changed by setting I/O allocations in the parameters.

(b) Q2ACPU-S1 system

System configuration	
Maximum number of extension stages	7th extension stage
Maximum number of I/O modules	64 modules
Maximum number of I/O points	1024 points
Main base/extension base types	A32B, A35B, A38B, A38HB/A62B, A65B, A68B, A52B, A55B, A58B
Extension cable types	AC06B, AC12B, AC30B
Restrictions	(1) It is not possible to connect an extension base unit to an A32B main base unit (there is no connector for extensions on the A32B). (2) When using an A52B, A55B, or A58B extension base unit, since 5 VDC is supplied from the power supply module on the main base unit, refer to Section 17.3 for details on use and to determine whether or not it is feasible to use these units. (3) The overall distance of the extension cable must not be longer than 6.6 m (21.65 ft).
Allocation of I/O numbers	(1) I/O numbers are allocated in accordance with the order of the numbers set for the extension base stages, regardless of the order in which the extension cables are connected. (2) I/O numbers are allocated on the assumption that all main base units and extension base units have 8 slots. Consequently, allocations of 16 points per slot are made for the parts of the system configuration drawing indicated with dotted lines. (3) Allocate 16 points to vacant slots. (4) Extension stage numbers do not need to be sequential, but any extension stage omitted from the system will occupy 8 (slots) x 16 I/O points. (5) With regard to items (2) to (4) above, the I/O allocations can be changed by setting I/O allocations in the parameters.

(c) Q3A/Q4ACPU system

System configuration	
Maximum number of extension stages	7th extension stage
Maximum number of I/O modules	64 modules
Maximum number of I/O points	A3U: 2048 A4U: 4096
Main base/extension base types	A32B, A35B, A38B, A38HB/A62B, A65B, A68B, A52B, A55B, A58B
Extension cable types	AC06B, AC12B, AC30B
Restrictions	(1) It is not possible to connect an extension base unit to an A32B main base unit (there is no connector for extensions on the A32B). (2) When using an A52B, A55B, or A58B extension base unit, since 5 VDC is supplied from the power supply module on the main base unit, refer to Section 17.3 for details on use and to determine whether or not it is feasible to use these units. (3) The overall distance of the extension cable must not be longer than 6.6 m (21.65 ft).
Allocation of I/O numbers	(1) I/O numbers are allocated in accordance with the order of the numbers set for the extension base stages, regardless of the order in which the extension cables are connected. (2) I/O numbers are allocated on the assumption that all main base units and extension base units have 8 slots. Consequently, allocations of 16 points per slot are made for the parts of the system configuration drawing indicated with dotted lines. (3) Allocate 16 points to vacant slots. (4) Extension stage numbers do not need to be sequential, but any extension stage omitted from the system will occupy 8 (slots) x 16 I/O points. (5) With regard to items (2) to (4) above, the I/O allocations can be changed by setting I/O allocations in the parameters.

3.3 System Equipment

3.3.1 System equipment list

The system equipment (modules and peripheral devices) that can be used in a QnACPU system are listed here.

(1) Modules for use with QnACPU

Module	Model	QnA Use Only	QnA/A Use	Description	Number of Occupied I/O Points (Module Type In I/O Allocation)	Current Consumption		Remark
						5 VDC (A)	24 VDC (A)	
CPU module	Q2ACPU	—	—	Number of I/O points: 512, internal RAM: 28 k steps	—	0.3	—	Memory card procured separately. Includes memory card current consumption.
	Q2ACPU-S1			Number of I/O points: 1024, internal RAM: 60 k steps		0.3	—	
	Q3ACPU			Number of I/O points: 2048, internal RAM: 92 k steps		0.3	—	
	Q4ACPU			Number of I/O points: 4096, internal RAM: 124 k steps		0.6	—	
Memory card	Q1MEM-64S	o	x	SRAM, 64 k bytes	—	—	—	
	Q1MEM-128S			SRAM, 128 k bytes				
	Q1MEM-256S			SRAM, 256 k bytes				
	Q1MEM-512S			SRAM, 512 k bytes				
	Q1MEM-1MS			SRAM, 1 M bytes				
	Q1MEM-2MS			SRAM, 2 M bytes				
	Q1MEM-64SE			SRAM, 32 k bytes; E^2PROM, 32 k bytes				
	Q1MEM-128SE			SRAM, 64 k bytes; E^2PROM, 64 k bytes				
	Q1MEM-256SE			SRAM, 128 k bytes; E^2PROM, 128 k bytes				
	Q1MEM-512SE			SRAM, 256 k bytes; E^2PROM, 256 k bytes				
	Q1MEM-1MSE			SRAM, 512 k bytes; E^2PROM, 512 k bytes				
	Q1MEM-256SF			SRAM, 128 k bytes; flash memory, 128 k bytes				
	Q1MEM-512SF			SRAM, 256 k bytes; flash memory, 256 k bytes				
	Q1MEM-1MSF			SRAM, 512 k bytes; flash memory, 512 k bytes				
	Q1MEM-2MSF			SRAM, 1 M bytes, flash memory, 1 M bytes				

Module	Model	QnA Use Only	QnA/A Use	Description	Number of Occupied I/O Points (Module Type in I/O Allocation)	Current Consumption		Remark
						5 VDC (A)	24 VDC (A)	
Input modules	AX10	x	o	16-input 100 VAC input module	16 (16 inputs)	0.055	—	
	AX11			32-input 100 VAC input module	32 (32 inputs)	0.11	—	
	AX20			16-input 200 VAC input module	16 (16 inputs)	0.055	—	
	AX21			32-input 200 VAC input module	32 (32 inputs)	0.11	—	
	AX31			32-input 12/24 V AC/DC input module	32 (32 inputs)	0.11	—	
	AX40			16-input 12/24 VDC input module	16 (16 inputs)	0.055	—	
	AX41			32-input 12/24 VDC input module	32 (32 inputs)	0.11	—	
	AX42			64-input 12/24 VDC input module	64 (64 inputs)	0.12	—	
	AX50			16-input 48 VDC sink input module	16 (16 inputs)	0.055	—	
	AX50-S1			16-input 48 VDC sink/source input module	16 (16 inputs)	0.055	—	
	AX60			16-input 100/110/125 VDC sink input module	16 (16 inputs)	0.055	—	
	AX60-S1			16-input 100/110/125 VDC sink/source input module	16 (16 inputs)	0.055	—	
	AX70			16-input input module for sensor	16 (16 inputs)	0.055	—	
	AX71			32-input input module for sensor	32 (32 inputs)	0.11	—	
	AX80			16-input 12/24 VDC source input module	16 (16 inputs)	0.055	—	
	AX80E			16-input 12/24 VDC source input module	16 (16 inputs)	0.055	—	
	AX81			32-input 12/24 VDC source input module	32 (32 inputs)	0.11	—	
	AX81-S2			32-input 48/60 VDC source input module	32 (32 inputs)	0.11	—	
	AX81B			32-input 24 VDC sink/source input module	64 (32 inputs)	0.125	—	
	AX82			64-point 12/24 VDC source input module	64 (64 inputs)	0.12	—	

REMARK

"QnA use only" modules are those that can only be used with QnACPU.
"QnA/A use" modules are those that can be used with either QnACPU or ACPU. (However, there are some restrictions when some of these modules are used with QnACPU).

Module	Model	QnA Use Only	QnA/A Use	Description	Number of Occupied I/O Points (Module Type in I/O Allocation)	Current Consumption		Remark
						5 VDC (A)	24 VDC (A)	
Output modules	AY10	x	o	16-output relay contact output module (2 A)	16 (16 outputs)	0.115	0.15	*1: Indicates a source load module. Other modules are sink load modules.
	AY10A			16-output relay contact output module, for independent contact output	16 (16 outputs)	0.115	0.15	
	AY11			16-output relay contact output module, with surge suppression	16 (16 outputs)	0.115	0.15	
	AY11A			16-output relay contact output module, for independent contact output, with surge suppression	16 (16 outputs)	0.115	0.15	
	AY11E			16-output relay contact output module (fused)	16 (16 outputs)	0.115	0.15	
	AY13			32-point relay contact output module (2 A)	32 (32 outputs)	0.23	0.29	
	AY13E			32-point relay contact output module (fused)	32 (32 outputs)	0.23	0.29	
	AY22			16-point triac output module (2 A, fused)	16 (16 outputs)	0.305	—	
	AY23			32-point triac output module (0.6 A, fused)	32 (32 outputs)	0.59	—	
	AY40			16-output 12/24 VDC transistor output module (0.1 A)	16 (16 outputs)	0.115	0.016	The short protection and overheat protection functions of the AY40P, AY41P, AY60EP, AY80EP, AY81EP, and AY82EP are described below:
	AY40A			16-output 12/24 VDC transistor output module, for independent contact output (0.3 A)	16 (16 outputs)	0.19	—	
	AY40P			16-output 12/24 VDC transistor output module, with short protection function and overheat protection function	16 (16 outputs)	0.115	0.03	
	AY41			32-output 12/24 VDC transistor output module (0.1 A)	32 (32 outputs)	0.23	0.04	Short protection function Function that protects the transistors from overcurrents occurring, for example, due to short circuits in external wiring.
	AY41P			32-output 12/24 VDC transistor output module, with short protection function and overheat protection function	32 (32 outputs)	0.23	0.06	
	AY42			64-output 12/24 VDC transistor output module (0.1 A)	64 (64 outputs)	0.29	0.08	
	AY42-S3			64-output 12/24 VDC transistor output module (fused)	64 (64 outputs)	0.29	0.08	
	AY42-S4			64-output 12/24 VDC transistor output module, with photocoupler with built-in Zener diode	64 (64 outputs)	0.50	—	Overheat protection function Function that protects the transistors from damage due to external temperature rise attributable to external causes.
	AY50			16-output 12/24 VDC transistor output module (0.5 A, fused)	16 (16 outputs)	0.115	0.13	
	AY51			32-output 12/24 VDC transistor output module (0.5 A)	32 (32 outputs)	0.23	0.10	
	AY51-S1			32-output 12/24 VDC transistor output module (0.3 A, fused)	32 (32 outputs)	0.31	0.02	
	AY60			16-output 12/24/48 VDC transistor output module (2 A, fused)	16 (16 outputs)	0.115	0.13	

Module	Model	QnA Use Only	QnA/A Use	Description	Number of Occupied I/O Points (Module Type in I/O Allocation)	Current Consumption		Remark
						5 VDC (A)	24 VDC (A)	
Output modules	*1 AY60E	x	o	16-output 12/24/48 VDC transistor output module (fused) 12/24 VDC: 2 A, 48 VDC: 0.8 A (fused)	16 (16 outputs)	0.115	0.13	*1: Indicates a source load module. Other modules are sink load modules. The short protection and overheat protection functions of the AY40P, AY41P, AY60EP, AY80EP, AY81EP, and AY82EP are described below: Short protection function Function that protects the transistors from overcurrents occurring, for example, due to short circuits in external wiring. Overheat protection function Function that protects the transistors from damage due to external temperature rise attributable to external causes.
	AY60S			16-output 12/24/48 VDC transistor output module (2 A)	16 (16 outputs)	0.075	0.006	
	*1 AY60EP			16-output 12/24 VDC transistor output module (2 A), with short protection function and overheat protection function	16 (16 outputs)	0.115	0.22	
	AY70			16-output, CMOS (5/12 VDC) output module (16 mA)	16 (16 outputs)	0.10	12 VDC 0.11	
	AY71			32-output, CMOS (5/12 VDC) output module (16 mA)	32 (32 outputs)	0.20	12 VDC 0.20	
	AY72			64-output, CMOS (5/12 VDC) output module (16 mA)	64 (64 outputs)	0.30	12 VDC 0.60	
	*1 AY80			16-output 12/24/48 VDC transistor output module (0.5 A, fused)	16 (16 outputs)	0.115	0.12	
	*1 AY80EP			16-output 12/24 VDC transistor output module (0.8 A), with short protection function and overheat protection function	16 (16 outputs)	0.115	0.22	
	*1 AY81			32-output 12/24 VDC transistor output module (0.5 A)	32 (32 outputs)	0.23	0.10	
	*1 AY81EP			32-output 12/24 VDC transistor output module (0.8 A), with short protection function and overheat protection function	32 (32 outputs)	0.23	0.44	
	*1 AY82EP			64-output 12/24 VDC transistor output module (0.1 A), with short protection function and overheat protection function	64 (64 outputs)	0.29	0.10	
Dynamic input/output combination module	A42XY	x	o	64-input, 64-output, dynamic scanning	64 (64 outputs)	0.11	0.235	Performs I/O processing in 8-point units independently of the CPU module, while scanning.
Input/output combination module	AH42	x	o	32-input, 32-output, 12/24 VDC transistor output module (0.1 A)	64 (64 outputs)	0.245	0.04	The first half 32 points are inputs and the second half 32 points are outputs.

	Module	Model	QnA Use Only	QnA/A Use	Description	Number of Occupied I/O Points (Module Type in I/O Allocation)	Current Consumption		Remark
							5 VDC (A)	24 VDC (A)	
Special function modules	Single-axis positioning module	AD70	x	o	For single-axis positioning control, speed control, and speed/position switching control. (0 to ±10 V)	32 (special 32 points)	0.3	—	
		AD70D			1-axis, digital output, for MR-SB(K)/SD	32 (special 32 points)	0.8	—	
	Positioning module	AD71	x	o	For positioning control. Pulse train output, 2-axes (independent control, simultaneous 2-axis control, linear interpolation control). When in combination with AD76, stepping motors can be used.	32 (special 32 points)	1.5	—	
		AD71S7							
		AD71S1			For positioning control. (Dedicated to MELDAS-S1 support driver.) Pulse train output, 2-axes (independent control, simultaneous 2-axis control, linear interpolation control).	32 (special 32 points)	1.5	—	
		AD71S2			For positioning control, for high-speed control. Pulse train output, 2-axes (independent control, simultaneous 2-axis control, linear interpolation control). When in combination with AD76, stepping motors can be used.	32 (special 32 points)	1.5	—	
		AD72			For positioning control. Analog voltage output (0 to ±10 V). 2-axes (independent control, simultaneous 2-axis control, linear interpolation control).	48 (first half: vacant 16 points second half: special 32 points)	0.9	—	
		AD76			Driver for stepping motor. Used in combination with AD71 or AD71S2.	16 (vacant 16 points)	—	—	
	Position detection module	A61LS	x	o	Absolute detection method. Resolution: One resolver revolution = 4096 divisions. Response speed: within 6 ms	48 (first half: special 32 points second half: vacant 16 points)	0.8	—	
		A62LS			Absolute detection method, multiple rotation type. Resolution: One resolver revolution = 4096 divisions. Response speed: 2 ms	48 (first half: special 32 points second half: vacant 16 points)	1.5	—	• The resolution depends on the connected resolver.
	High-speed counter module	AD61	x	o	24-bit binary, 1/2 phase input, reversible counter, 50 kPPS, 2 channels	32 (special 32 points)	0.3	—	
		AD61S1			24-bit binary, 1/2 phase input, reversible counter, 1 phase ... 10 kPPS, 2 phases ... 7 kPPS, 2 channels	32 (special 32 points)	0.3	—	

Module		Model	QnA Use Only	QnA/A Use	Description	Number of Occupied I/O Points (Module Type in I/O Allocation)	Current Consumption		Remark
							5 VDC (A)	24 VDC (A)	
Special function modules	A-D converter module	A68AD	x	o	4 to 20 mA / 0 to ±10 V Analog input, 8 channels	32 (special 32 points)	0.9	—	
		A68AD-S2							
		A68ADN			0 to ±20 mA / 0 to ±10 V Analog input, 8 channels	32 (special 32 points)	0.4	—	
		A616AD			4 to 20 mA / 0 to ±10 V Analog input, 16 channels. Expansion to maximum of 121 channels possible by using A60MX(R)	32 (special 32 points)	1.0	—	
		A60MX			Multiplex module (IC relay) Analog input, 16 channels	16 (vacant 16 points)	0.65	—	Used in combination with A616AD or A616TD.
		A60MXR			Multiplex module (mercury relay). Analog input, 16 channels	16 (vacant 16 points)	0.5	—	
	Temperature-digital converter module	A616TD	x	o	For temperature detection by thermocouple (when connected to A60MXT). 0 to ±10 V / 0 to 20 mA (when connected to A60MX(R))	32 (special 32 points)	1.0	—	
		A60MXT			Multiplex module. Temperature input: 15 channels. Temperature detection by thermocouple when used in conjunction with A616TD.	32 (first half: vacant16 points second half: vacant16 points)	0.8	—	Used in combination with A616TD.
		A68RD3			−180 to 600 °C temperature input module (For 3-wire type platinum resistor)	32 (special 32 points)	0.94	—	
		A68RD4			−180 to 600 °C temperature input module (For 4-wire type platinum resistor)		0.75	—	
	D-A converter module	A68DAV	x	o	0 to ±10 V, analog output, 8 channels.	32 (special 32 points)	0.15	0.2	
		A68DAI			0 to 20 mA, analog output, 8 channels.	32 (special 32 points)	0.15	0.4	
		A62DA			4 to 20 mA / 0 to ±10 V Analog output 12 bits, 2 channels	32 (special 32 points)	0.6	0.35	
		A62DA-S1			4 to 20 mA / 0 to ±10 V Analog output, 2 channels				
		A616DAI			4 to 20 mA. Resolution: 1/4000 Analog output, 16 channels	32 (special 32 points)	0.3	—	15 VDC (A68P) is required. +0.53 A −0.125 A
		A616DAV			0 to ±10 V/0 to ±5 V. Resolution: 1/4000 Analog output, 16 channels	32 (special 32 points)	0.38	—	15 VDC (A68P) is required. +0.2 A −0.17 A
	A-D, D-A converter module	A84AD	x	o	4 to 20 mA / 0 to ±10 V Analog input/output 15 bits, 4 channels	48 (first half: vacant 16 points second half: special 32 points)	0.24	0.53	
	CRT control module	AD57	x	o	CRT display, semigraphic Selection between color/monochrome possible.	64 (special 64 points)	1.21	0.16	To create canvas and character generator ROMs, use SW□GP-AD57P.
		AD57S1					1.55		

Module		Model	QnA Use Only	QnA/A Use	Description	Number of Occupied I/O Points (Module Type in I/O Allocation)	Current Consumption		Remark
							5 VDC (A)	24 VDC (A)	
Special function modules	Memory card, Centronics interface module	AD59	x	o	32 k byte memory battery backup. Can be connected to printer conforming to Centronics standards	32 (special 32 points)	0.3	—	
		AD59-S1					0.32		0.35A when connected to AD59MEF.
	Voice output module	A11VC	x	o	Messages can be recorded and played back on a maximum of 60 channels. The following recording times can be selected for each channel: 1 second, 2 seconds, 4 seconds, 8 seconds. The total recording time is 64 seconds	16 (special 16 points)	0.6	0.38	
Network module		AJ71QLP21	o	x	For MELSECNET/10 optical loop networks	32 (special 32 points)	0.65	—	Up to 4 modules can be used with one CPU.
		AJ71QLP21S			For MELSECNET/10 optical loop networks. Network backup by external power supply	32 (special 32 points)	0.65	0.2	
		AJ71QBR11			For MELSECNET/10 coaxial bus networks	32 (special 32 points)	0.8	—	
		AJ71QLP25			For MELSECNET/10 optical loop network remote I/O stations	—	0.8	—	
		AJ71QBR15			For MELSECNET/10 coaxial bus network remote I/O stations	—	0.9	—	
Data link module		*2 AJ71AP21	x	o	For MELSECNET II optical data links	32 (special 32 points)	0.5	—	Up to 2 modules can be used with one CPU.
		*2 AJ71AR21			For MELSECNET II coaxial data links	32 (special 32 points)	0.9	—	
		*2 AJ71AT21B			For MELSECNET/B data links	32 (special 32 points)	0.72	—	
		*2 AJ72T25B			For MELSECNET/B data link remote I/O stations	—	0.3	—	
Serial communications module		AJ71QC24	o	x	Link module that communicates data with a computer. Transmission speed: 300 bps to 19.2 kbps RS-232, RS-422/485: one channel each	32 (special 32 points)	0.3	—	
		AJ71QC24-R2			Link module that communicates data with a computer. Transmission speed: 300 bps to 19.2 kbps RS-232C: two channels	32 (special 32 points)	0.2	—	
		AJ71QC24-R4			Link module that communicates data with a computer. Transmission speed: 300 bps to 19.2 kbps RS-422: one channel each	32 (special 32 points)	0.38	—	

Module	Model	QnA Use Only	QnA/A Use	Description	Number of Occupied I/O Points (Module Type in I/O Allocation)	Current Consumption		Remark
						5 VDC (A)	24 VDC (A)	
Computer link module (for AnUCPU)	*2 AJ71UC24	x	o	Link module that communicates data with a computer. Transmission speed: 300 bps to 19.2 kbps RS-232C, RS-422: one channel each, compatible with RS485	32 (special 32 points)	1.4	—	A total of up to 6 modules can be used with one CPU. *4 When power supply equipment is not connected. 0.7 A
Computer link module	*3 AJ71C24(S3)	x	o	Link module that communicates data with a computer. Transmission speed: 300 bps to 19.2 kbps RS-232C, RS422: one channel each	32 (special 32 points)	1.4	—	
	*2 AJ71C24-S6/S8							
Intelligent communication module	*3 AD51-S3	x	o	GPC-BASIC, maximum of 8 tasks. RS-232C, RS-422: two channels each available	48 (first half: vacant 16 points second half: special 32 points)	1.3	—	
	*2 AD51H-S3	x	o	AD51H-BASIC, maximum of 8 tasks. Equipped with IC memory card interface				
External fault diagnosis module	*2 AD51FD	x	o	6 types of fault detection. Can output alarms and fault diagnosis data to external destinations.	48 (first half: vacant 16 points second half: special 32 points)	1.3	—	
Ethernet interface module	*2 AJ71E71	x	o	10BASE5/10BASE2 specifications. Transmission speed: 10 Mbps	32 (special 32 points)	1.5	—	
SUMINET interface module	*3 AJ71P41	x	o	For SUMINET-3200 networks. Communication speed: 2 Mbps	32 (special 32 points)	*4 0.4	—	
Host controller high-speed link module	*2 AJ71C23-S3	x	o	Link module that sends/receive data at high speed to/from a computer. Transmission speed: 500 kbps RS-422: one channel each	32 (special 32 points)	1.5	—	
Terminal interface module	AJ71C21	x	o	Link module that communicates data using a BASIC function terminal interface or the no-protocol mode. Transmission speed: 600 bps to 19.2 kbps RS-232C, RS-422: one channel each	32 (special 32 points)	0.8	—	
	AJ71C21S1							
Multidrop data link module	AJ71C22S1	x	o	Sends and receives bit data to a maximum of 8 slave stations to which it is connected in a multidrop system. Used for the master station of a multidrop link. Transmission speed: 38.4 kbps RS-422: one channel each	32 (special 32 points)	1.4	—	
Multidrop data link module	A0J2C25	x	o	Used as a remote I/O station of a multidrop link.	—	—	—	
	A0J2C214			Used as a local station in a multidrop link. (In A0J2CPU and A0J2HCPU systems, can also be used as the master station in computer links and multidropdata links)	64 points	0.3	—	

Module		Model	QnA Use Only	QnA/A Use	Description	Number of Occupied I/O Points (Module Type in I/O Allocation)	Current Consumption		Remark
							5 VDC (A)	24 VDC (A)	
Graphic operation terminal		*2 A64GOT-L A64GOT-LT21B	x	o	Compact graphic operation terminal. Integral monitor (monochrome liquid crystal). Number of monitor screens: 250, number of parts: 255	—	—	0.4	
		*2 A77GOT-S5			Large scale graphic operation terminal. Integral monitor (monochrome liquid crystal, color liquid crystal, EL). Number of monitor screens: 250, number of parts: 255	—	—	—	When a bus connection is made, the number of occupied points is 32 (special). Power consumption 100 VA
MELSECNET /MINI-S3 data link module		AJ71PT32-S3	x	o	Used for remote I/O control and remote terminal control with a total of 512 I/O points, for a maximum of 64 MELSECNET/MINI-S3 master stations.	I/O dedicated mode: 32 (special 32 points)	0.34	—	
		AJ71T32-S3				Extension mode: 48 (special 48 points)			
B/NET interface module		AJ71B62-S3	x	o	Used for B/NET transmission terminal control. Up to 63 stations can be controlled per module.	32 (special 32 points)	0.17	—	
Interrupt module		AI61	x	o	Used to designate execution of interrupt programs (16 interrupt inputs).	32 (special 32 points)	0.14	—	Only one module can be used per CPU.
Dummy module		AG62	x	o	Module allows selection of 16, 32, 48, or 64 points.	Setting range (Set number of points / inputs)	0.07	—	Has 16 simulation switches.
Blank cover		AG60	x	o	Keeps unused slots free of dust.	16 (vacant 16 points)	—	—	
Power supply module	Mounting position: power supply slot	A61P	x	o	Input: 100/200 VAC Output: 5 VDC 8 A		—	—	
		A61PEU							
		A62P			Input: 100/200 VAC Output: 5 VDC 5 A, 24 VDC 0.8 A				
		A62PEU							
		A63P			Input: 24 VDC Output: 5 VDC 8 A				
		A65P			Input: 100/200 VAC Output: 5 VDC 2 A, 24 VDC 1.5 A				Cannot be used with a main base unit.
		A67P			Input: 110 VDC Output: 5 VDC 8 A				
	Mounting position: I/O slot	A66P			Input: 100/200 VAC Output: 24 VDC 1.2 A	16 (vacant 16 points)			Power supply for AD70, A616DAV, A616DAI
		A68P			Input: 100/200 VAC Output: +15 VDC 1.2 A, −15 VDC 0.7 A	32 (first half: vacant 16 points second half: vacant 16 points)			

Module		Model	QnA Use Only	QnA/A Use	Description		Number of Occupied I/O Points (Module Type in I/O Allocation)	Current Consumption 5 VDC (A)	Current Consumption 24 VDC (A)	Remark
Base unit	Main base unit	A38HB	o	x	Can accommodate 8 I/O modules.		—	—	—	For high-speed access (dedicated to QnACPU)
		A38B	x	o	Can accommodate 8 I/O modules.					
		A35B			Can accommodate 5 I/O modules.					
		A32B			Can accommodate 2 I/O modules.					No connector for extension.
		A32B-S1			Can accommodate 2 I/O modules.					Has connector for extension.
	Extension base unit	A68B			Can accommodate 8 I/O modules.		—	—	—	Power supply module required.
		A65B			Can accommodate 5 I/O modules.					
		A62B			Can accommodate 2 I/O modules.					
		A58B			Can accommodate 8 I/O modules.		—	—	—	A61P, A62P, A63P, and A65P cannot be mounted.
		A55B			Can accommodate 5 I/O modules.					
		A52B			Can accommodate 2 I/O modules.					
Extension cable		AC06B	x	o	600 mm (23.62 in) long	Cables for connections between base units	—	—	—	
		AC12B			1200 mm (47.24 in) long					
		AC30B			3000 mm (118.11 in)long					
Simulation switch		A6SW16	x	o	16-point simulation switch		—	—	—	Installed in an input module.
		A6SW32			32-point simulation switch					
Miscellaneous	Battery	A6BAT	x	o	IC-RAM memory backup		—	—	—	
	Fuses: For AY11E, AY13E	MF51NM8			Cartridge type, 8 A		—	—	—	
	Fuses: For AY22	HP-70K			Plug type, 7 A					
	Fuses: For AY23	HP-32			Plug type, 3.2 A					
	Fuses: For AY50, AY80	MP-20			Plug type, 2 A					
	Fuses: For AY60	MP-32			Plug type, 3.2 A					
	Fuses: For AY60E	MP-50			Plug type, 5 A					
	Fuses: For power supply	GTH4			Cartridge type, 4 A					
	Fuses: For A63P	SM6.3A			Cartridge type, 6.3 A					

*2: Only internal devices within the AnACPU range can be accessed (file registers cannot be accessed).

*3: Only internal devices within the A3HCPU range can be accessed (file registers cannot be accessed).

Module	Model	Description	Applicable Models
Battery	A6BAT	IC-RAM memory backup	Installed in QnACPU module
Connector/terminal block converter module	A6TBXY36	For sink type input modules and sink type output modules (standard type)	AX42(S1), AY42(S1/S3/S4), AH42
	A6TBXY54	For sink type input modules and sink type output modules (2-wire type)	
	A6TBX70	For sink type input modules (3-wire type)	AX42(S1), AH42
	A6TBX36-E	For source type input modules (standard type)	AX82
	A6TBY36-E	For source type output modules (standard type)	AY82EP
	A6TBX54-E	For source type input modules (2-wire type)	AX82
	A6TBY54-E	For source type output modules (2-wire type)	AY82EP
	A6TBX70-E	For source type input modules (3-wire type)	AX82
Cable for connector /terminal block converter module	AC05TB	0.5 m (19.69 in), for source module	A6TBXY36 A6TBXY54 A6TBX70
	AC10TB	1 m (39.37 in), for source module	
	AC20TB	2 m (78.74 in), for source module	
	AC30TB	3 m (118.11 in), for source module	
	AC50TB	5 m (96.85 in), for source module	
	AC05TB-E	0.5 m (19.69 in), for source module	A6TBX36-E A6TBY36-E A6TBX54-E A6TBY54-E A6TBX70-E
	AC10TB-E	1 m (39.37 in), for source module	
	AC20TB-E	2 m (78.74 in), for source module	
	AC30TB-E	3 m (118.11 in), for source module	
	AC50TB-E	5 m (96.85 in), for source module	
Relay terminal module	A6TE2-16SR	For sink type output module	AY42, AY42-S1, AY42-S3, AY42-S4, AH42
Cable for connecting relay terminal module	AC06TE	0.6 m (23.62 in) long	A6TE2-16SR
	AC10TE	1 m (39.37 in) long	
	AC30TE	3 m (118.11 in) long	
	AC50TE	5 m (96.85 in) long	
	AC100TE	10 m (393.7 in) long	

(2) Peripheral devices

Device	Model	Remark
Programming unit	Q6PU	Connected to the CPU module by an RS-422 cable (AC30R4-PUS, AC20R4-A8PU); for program writing and reading. (5 VDC 0.4 A)
RS-422 cable	AC30R4-PUS	Cable for connection between CPU module and Q6PU. 3 m (118.11 in) long
	AC20R4-A8PU	Cable for connection between CPU module and Q6PU. 2 m (78.74 in) long

3.3.2 Cautions on system configurations

Details on the hardware and software that can be used with the QnACPU are presented below.

(1) Hardware

(a) Depending on the module type, there are restrictions on the number of modules that can be mounted, as shown below.

Module Type	For Use with QnACPU Only	For Use with ACPU	Remark
I/O module	—	No restrictions	—
Special function modules	No restrictions	No restrictions	—
Intelligent special function modules	No restrictions	Total of 6	—
Interrupt module	—	One only	—
Link modules	Total of 4 for network use	Total of 4 for network use Total of 2 for data link use	Total of 4 for network and data link use

REMARK

The modules tabled above are categorized as follows.

(1) I/O module ..... Standard input modules and output modules.

(2) Special function modules ..... Special function modules that perform processing in accordance with FROM/TO instructions from the QnACPU (for example: A68AD, A62DA, etc.)

(3) Intelligent special function modules .. Special function modules that, in addition to processing in accordance with FROM/TO instructions from the QnACPU, can also access the QnACPU (for example: AJ71UC24, AJ71QC24, etc.)

(4) Interrupt module ..... Modules that issue interrupts to the QnACPU (AI61)

(5) Link modules ..... Special function modules for MELSECNET II, /B data links and MELSECNET/10 networks.

The following special function modules cannot be used with QnACPU:

- AJ71C23 (host controller high-speed link module)
- AD57-S2 (A6MD controller module)
- AJ71C24 (computer link module) ..... Applies to products manufactured up to February 1987 only
  (Products manufactured from March 1987 onward, and products for which "H" is displayed for DATE (corresponding to A3H) can be used.)
- AD51 (intelligent communication module) .... Applies to products manufactured up to March 1987 only
  (Products manufactured from March 1987 onward, and products for which "H" is displayed for DATE (corresponding to A3H) can be used.)

- AJ71LP21, AJ71BR11 (MELSECNET/10 network modules)
- AJ71LP25, AJ71BR15 (MELSECNET/10 remote I/O modules)

(b) When a special function module is used with QnACPU, the range of devices that can be used is determined by the model of special function module.

Model / Device	AD51(S3), AJ71C24-S3, AJ71P41	AD51H(S3), AD51FD-S3, AJ71C23-S3, AJ71C24-S6/S8, AJ71UC24, AJ71E71
	Only the device range equivalent to that of A3HCPU can be accessed. Reading/Writing of file registers, programs, etc. is not possible.	Only the device range equivalent to that of AnACPU can be accessed. Reading/Writing of file registers, programs, etc. is not possible.
I/O devices (X/Y)	X/Y0 to 7FF	X/Y0 to 7FF
Internal relays (M, L, S)	M/L/S0 to 2047	M/L/S0 to 8191
Link relays (B)	B0 to 3FF	B0 to FFF
Timers (T)	T0 to 255	T0 to 2047
Counters (C)	C0 to 255	C0 to 1023
Data registers (D)	D0 to 1023	D0 to 8191
Link registers (W)	W0 to 3FF	W0 to FFF
Annunciators (F)	F0 to 255	F0 to 2047

(c) When a QnACPU is mounted on an A38HB high-speed access main base unit, reading and writing to and from the buffer memories of special function modules, intelligent special function modules and link modules can be performed at greater speeds.

(d) The following methods are used to connect graphic operation terminal units to a QnACPU.

<table>
<tr><th>Model</th><th>Connection Method</th><th>Accessible Device Range</th></tr>
<tr><td>A64GOT-L</td><td>Direct connection to CPU</td><td rowspan="4">The table below shows the ranges when a device range equivalent to that of AnACPU only can be accessed.
(File register reading/writing, system monitoring, ladder monitoring, etc., are not possible.)
<table><tr><td>Input/output devices</td><td>X/Y0 to 7FF</td></tr><tr><td>Internal relays</td><td>M/L/S0 to 8191</td></tr><tr><td>Link relays</td><td>B0 to FFF</td></tr><tr><td>Timers</td><td>T0 to 2047</td></tr><tr><td>Counters</td><td>C0 to 1023</td></tr><tr><td>Data registers</td><td>D0 to 8191</td></tr><tr><td>Link registers</td><td>W0 to FFF</td></tr><tr><td>Annunciators</td><td>F0 to 2047</td></tr></table></td></tr>
<tr><td>A64GOT-LT21B</td><td>Direct connection to CPU
MELSECNET/B connection</td></tr>
<tr><td>A77GOT</td><td>Direct connection to CPU
MELSECNET (II) connection
MELSECNET/B connection</td></tr>
<tr><td>A77GOT-S3</td><td>Direct connection to CPU
MELSECNET (II) connection
MELSECNET/B connection
MELSECNET/10 connection</td></tr>
<tr><td>A77GOT-S5</td><td>Direct connection to CPU
MELSECNET (II) connection
MELSECNET/B connection
MELSECNET/10 connection
Bus connection</td><td></td></tr>
</table>

(e) The accessible range for an AJ71UC24 computer link module comprises the CPU to which the AJ71UCPU is mounted (the host station) and the other stations in the network to which the host station is connected.

It is not possible to access other stations in other networks by using the MELSECNET/10 network system routing function.

The access range for an AJ71QC24 serial communication module is the host station, other stations in the network connected to the host station, and other stations in other networks accessed through up to 7 relay stations by using the routing function.

(2) Software packages

The system startup software packages that can be used to create programs for QnACPU are indicated below.

Peripheral Device Capable of GPP Functions	Software Package for System Startup
IBM PC/AT	SW□IVD-GPPQ

Apart from the above, the following software packages can be used.

- CAD interface package SW0IVD-CADQ
- Data conversion package SW0IVD-CNVQ
- Macro/library package SW0IVD-MSDQ
  SW0IVD-MSPQ
- Ladder sequence linking package SW0IVD-LNKQ

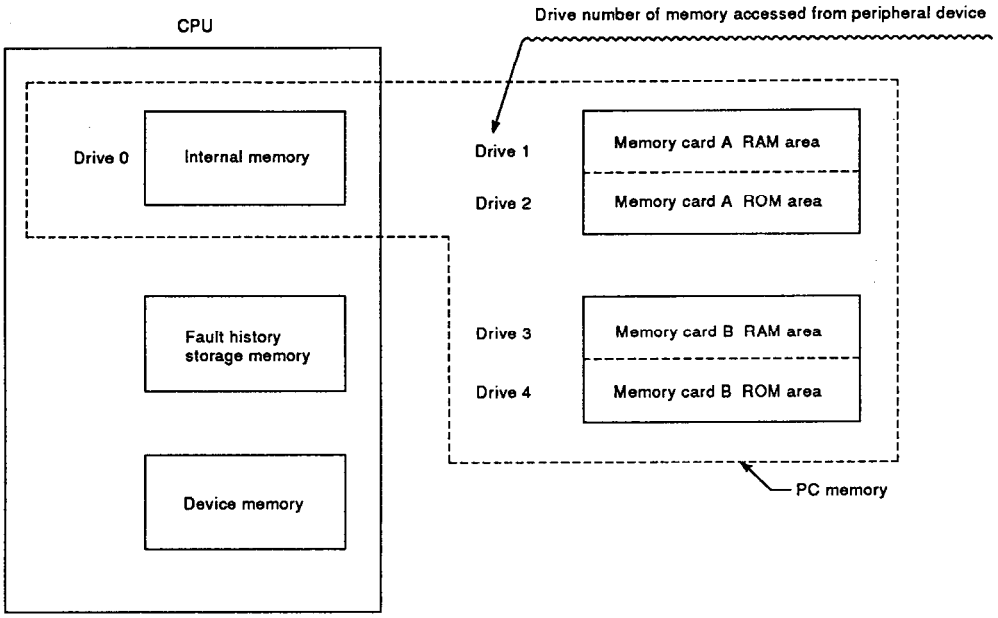
REMARK

The peripheral devices and software packages that can be used with QnACPU are listed below:

- A□PU (programming unit)
- A6WU (ROM writer unit)
- A6DU-B (data access unit)
- A6TEL (modem interface unit)
- A6GPP (intelligent GPP)
- A6HGP (handy graphic programmer)
- A6PHP (plasma handy graphic programmer)
- System startup software package for ACPU
  SW□□□-GPPA, SW□□□-SAP2

3.3.3 QnACPU memory block diagram

The following block diagram shows the QnACPU memory configuration.



- Internal memory : Memory that stores parameters, sequence programs, etc.
- Fault history storage memory : Memory that stores fault history data
- Device memory : Memory that stores device data
- Memory card (RAM, ROM area) : Memory that stores the files, comments, etc., for parameters, sequence programs, sampling traces, etc.
- PC memory : Indicates all the memories of drives 0 through 4.

4. PERFORMANCE SPECIFICATIONS

This section gives the performance specifications of the QnACPU.

Item			Model Name				Remark
			Q2ACPU	Q2ACPU-S1	Q3ACPU	Q4ACPU	
Control system			Repeated operation (using stored program)				
I/O control method			Refresh method				Direct input using device names possible
Programming language			Language dedicated to sequence control				
			Relay symbol language, logic symbolic language, MELSAP-3 (SFC)				
Processing speed (sequence instruction) (μs/step)		LD	0.2		0.15	0.075	
		MOV	0.6		0.45	0.225	
Number of instructions		Sequence instructions	39				
		Basic instructions	230				
		Application instructions	321				
		Dedicated instructions	171				
Constant scan (ms) (program started at fixed time intervals)			5 to 2000 (can be set in 5 ms units)				Set by parameter
Memory capacity			Capacity of the installed memory card (max. 2036 k bytes)				
Program capacity		Number of steps (steps)	Max. 28 k	Max. 60 k	Max. 92 k	Max. 124 k	
		Number of files (files)	28	60	92	124	
Number of I/O device points			8192 (X/Y0 to 1FFF)				Number of points that can be used in programs
Number of I/O points			512 (X/Y0 to 1FF)	1024 (X/Y0 to 3FF)	2048 (X/Y0 to 7FF)	4096 (X/Y0 to FFF)	Number of points actually accessible with I/O modules
Number of device points	Internal relays [M] (points)		Default: 8192 (M0 to 8191)				The number of devices is set by parameter
	Latch relays [L] (points)		Default: 8192 (L0 to 8191)				
	Link relays [B] (points)		Default: 8192 (B0 to 1FFF)				
	Timers [T] (points)		Default: 2048 (T0 to 2047) (Low-speed/high-speed used in conjection) Low-speed/high-speed switching set with instructions Low-speed/high-speed measurement units set by parameter (Low-speed: 10 to 1000 ms, 10 ms units, default: 100 ms) (High-speed: 1 to 100 ms, 1 ms units, default: 10 ms)				
	Retentive timers [ST] (points)		Default: 0 (ST0 to 2047)				

Item		Model Name				Remark
		Q2ACPU	Q2ACPU-S1	Q3ACPU	Q4ACPU	
Number of device points	Counters [C] (points)	• Ordinary counters Default: 1024 (C0 to 1023) • Interrupt pointers Max. 48 (parameter default: 0 points, set by)				The number of devices is set by parameter
	Data registers [D] (points)	Default: 12288 (D0 to 12287)				
	Link registers [W] (points)	Default: 8192 (W0 to 1FFF)				
	Annunciators [F] (points)	Default: 2048 (F0 to 2047)				
	Edge relays [V] (points)	Default: 2048 (V0 to 2047)				
	Special link relays [SB] (points)	Default: 2048 (SB0 to 7FF)				
	Special link registers [SW] (points)	Default: 2048 (SW0 to 7FF)				
	File registers [R] (points)	32768 (R0 to 32767) Up to 1042432 points can be used by block switching.				
		1042432 (ZR0 to 1042431) Block switching is not necessary.				
	Step relays [S] (points)	8192 (S0 to 8191)				The number of devices is fixed
	Index registers [Z] (points)	16 (Z0 to 15)				
	Pointers [P] (points)	4096 (P0 to 4095), ranges for pointers in files and common pointers set by parameter.				
	Interrupt pointers [I] (points)	48 (I0 to 47) The cycle interval for system interrupt pointers I28 to I31 is set by parameter (1 to 1000 ms, in 5 ms units)				
	Special relays [SM] (points)	2048(SM0 to 2047)				
	Special registers [SD] (points)	2048(SD0 to 2047)				
	Input bit condition devices [FX] for subroutine calls with attributes (points)	16(FX0 to 15)				
	Output bit condition devices [FY] for subroutine calls with attributes (points)	16(FY0 to 15)				
	Input/output data condition devices [FD] for subroutine calls with attributes (points)	5(FD0 to 4)				
Link direct devices		Devices that access link devices directly. For MELSECNET/10 only. Designation format: J□□\□□				
Special function module direct devices		Devices that directly access the buffer memories of special function modules. Designation format: U□□\G□□				
Latch (memory back up) range		L0 to 8191 (default) (Latch ranges can be set for B, F, V, T, ST, C, D, W devices.)				Set by parameter
Remote RUN/PAUSE contacts		One RUN contact and one PAUSE contact can be set from X0 to 1FFF				

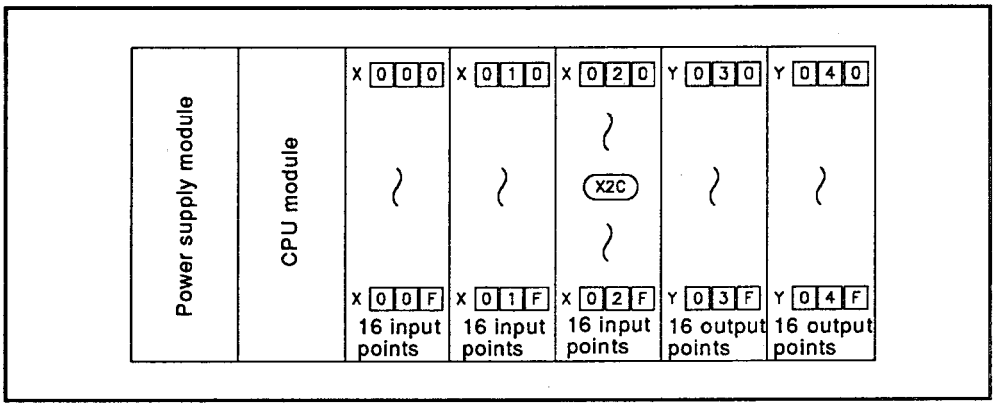
Item	Model Name				Remark
	Q2ACPU	Q2ACPU-S1	Q3ACPU	Q4ACPU	
Clock function	Year, month, date, hour, minute, second, day of week (automatic recognition of leap years) Accuracy –2.3 to + 4.4 s(TYP. +1.8 s) /d at 0 °C Accuracy –1.1 to + 4.4 s(TYP. +2.2 s) /d at 25 °C Accuracy –9.6 to + 2.7 s(TYP. +2.4 s) /d at 55 °C				
Allowable momentary power interruption time	Depends on the power supply module				See Section 16.1
Internal current consumption for 5 VDC (A)	0.3	0.3	0.3	0.6	
Weight kg (lb)	0.8 (1.76)	0.8 (1.76)	0.8 (1.76)	0.8 (1.76)	
External dimensions mm(inch)	250 (9.84) x 79.5 (3.13) x 121 (4.76)				

5. I/O NUMBER ALLOCATION

This section explains how to make the I/O number allocations at the QnACPU to enable data exchanges with I/O modules and special function modules.

5.1 About I/O Numbers

I/O numbers are used in the sequence program to input data from input modules and output data to output modules.
I/O numbers are expressed as three-digit hexadecimal numbers.
The I/O numbers when all the I/O modules occupy 16 points are indicated below.



Concept for I/O Numbers

REMARK

When programming with a peripheral device, I/O numbers can be input as 2 digits.

I/O number Input at peripheral device

X010 → X10

Y020 → Y20

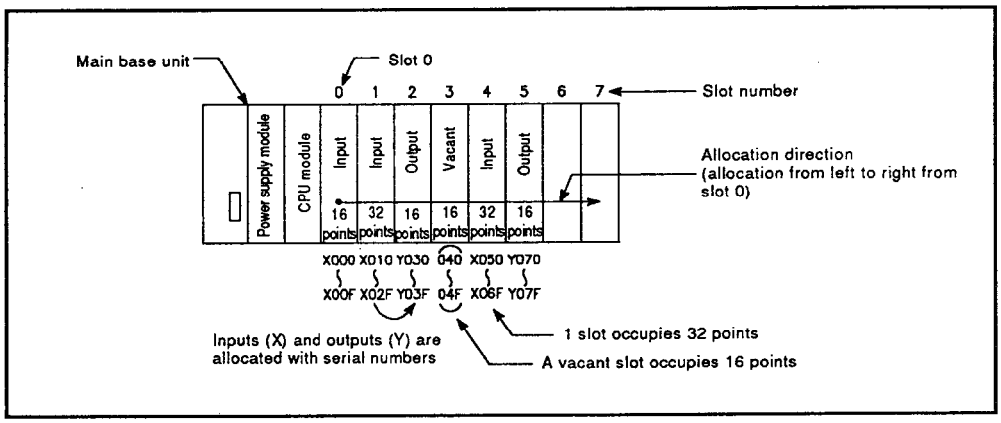
5.2 Basics of I/O Number Allocation

At power ON or reset, the I/O allocation described below is performed. In the sequence program, designate the I/O numbers allocated in accordance with the following considerations.

(1) I/O numbers are allocated sequentially from left to right, taking slot 0 (the slot to the right of the CPU module) of the main base unit to be "0".

(2) The I/O modules and special function modules mounted to the main base unit occupy the I/O numbers corresponding to the number of I/O points of the module in question.

(3) Vacant slots where no I/O module or special function module is loaded are allocated 16 points.

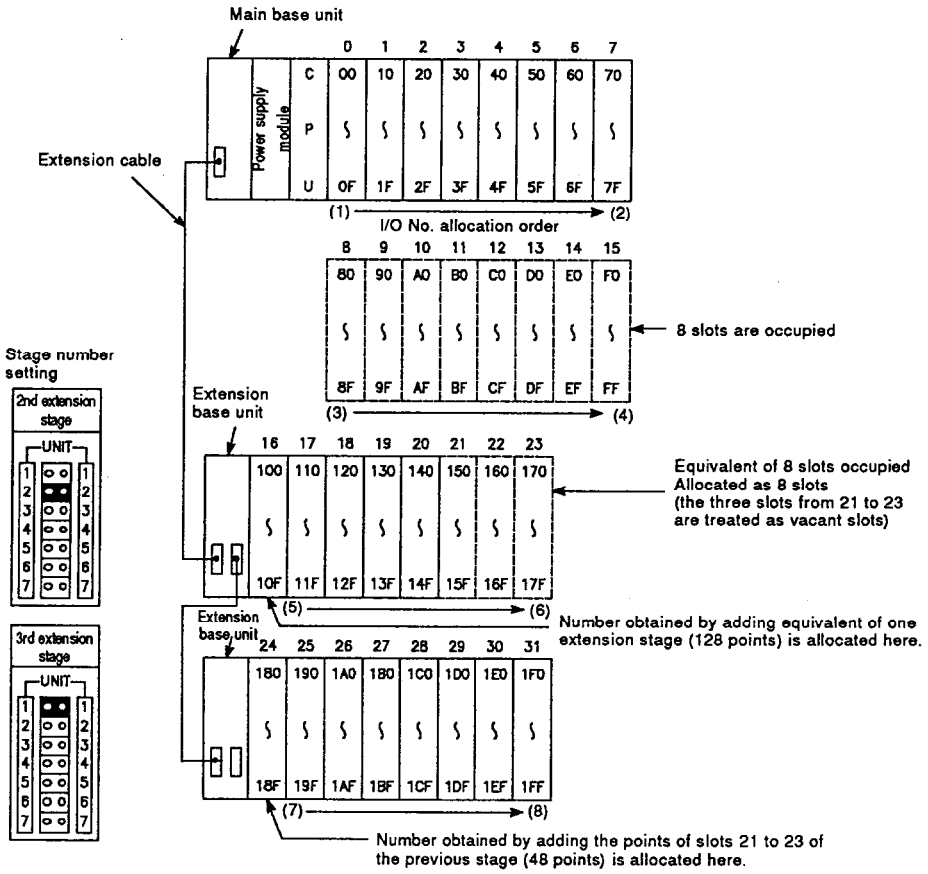


(4) If an extension base unit is connected, its allocation starts from the number immediately following the main base unit allocations.

The order of allocation for extension base units is not the order in which the extension base units are connected, but the order of the stage numbers set for the extension base units. It is possible to connect extension base units as follows: main base unit → 2nd extension base unit → 1st extension base unit.

(5) If any extension stage number is skipped, I/O allocations are made assuming that each of the 8 slots of the skipped extension stage occupy 16 points.

(6) I/O allocations are made assuming that every base unit has 8 slots. If a 5-slot base unit is used, an I/O number obtained by adding points equivalent to 3 slots (48 points) to the final I/O number of the 5-slot base unit is allocated to the next extension base unit.



* I/O numbers are allocated in the order (1) → (2) → (3) → (4) → (5) → (6) → (7) → (8)
The I/O allocations shown here assume that all modules are 16-point modules.

5.3 I/O Allocations Using GPPQ

When using a QnACPU, input/output modules and special function modules can be controlled even if I/O allocation is not performed with GPPQ.
I/O allocations made with GPPQ are valid in the following cases.

(1) Reasons for making I/O allocations with GPPQ

(a) To set 0 point for the equivalent of 3 slots when a 5-slot base unit is used.

(b) To make a reservation to change to a module other than a 16-point module for the purpose of system expansion.

(c) To prevent the I/O numbers from changing if an I/O module or special function module that occupies a number of points other than 16 has to be removed because it is faulty.

(d) To avoid I/O number amendments in programs by changing the I/O numbers allocated to each module on the base unit to match the I/O numbers of the program.

(2) Basics of I/O allocation using GPPQ

There are the following two methods for making I/O allocations using GPPQ.

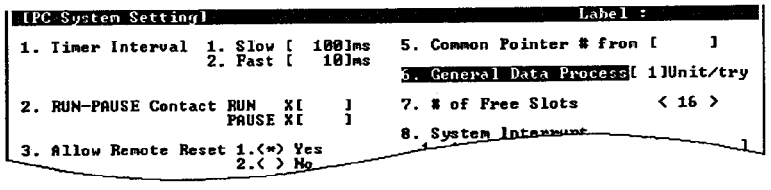
1) Set the number of points of the main base and extension base slots that are vacant (number of vacant slot points).

2) Set the I/O allocations for each module type in main base and extension base slot units (I/O allocation).

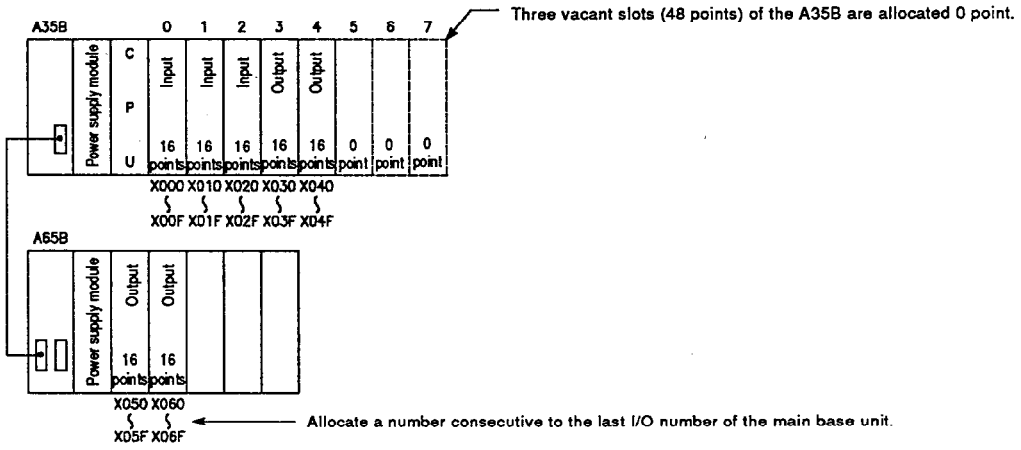
Parameter settings are used for both of these methods. If both methods 1) and 2) are set, the setting for 2) takes priority.

(a) Setting the number of vacant slot points

Set the number of points for all slots that are vacant on the base units.
In systems in which this setting is not made in the parameters, 16 points are set for vacant slots.
This setting is made using "7. # of Free Slots" on the "PC System Setting" screen in the parameter mode.

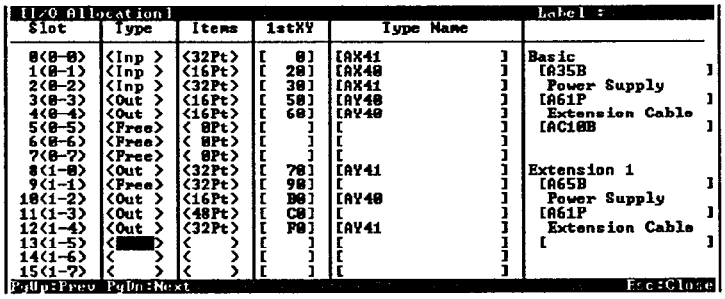


The setting is made in 16-point units within the range 0 to 64. The default value is 16 points.
Example: Assuming the number of vacant slot points is 0 points:



(b) Setting I/O allocations

The setting for the I/O allocation for each type of module in main base and extension base slot units is described here.
This setting is performed on the "I/O Allocation" screen in the parameter mode.



The setting details are as follows.

Item	Setting Details	Setting Range	Default Value
Slot settings	Set data for each slot (not necessary to make all settings).	Vacant/input/output /special	No setting
Type	Set the type of the module.		
Items	Set the number of points for the module.	0 to 64 points (16-point units)	
1stXY	Set the head number of the module's XY devices.	0 to 1FFF (16-point units)	
Type name	Set the model name of the module. *1	16 characters or fewer	
Base designation *2	Set data for each base unit.	16 characters or fewer	No setting
Power supply	Set the model name of the power supply module.		
Extension cable	Set the model name of the extension cable.	16 characters or fewer	

Skipped settings are handled as follows:

- Type and Number: In accordance with the loaded module.
- Head XY : The number following the total obtained by summing the numbers of points of the modules already set.
  If there is any duplication, an error (SP.UNIT LAY ERROR) occurs.

POINT

The power supply module name set in base unit designation is only used for the current capacity check in the PC diagnosis mode. It is not used by the CPU and it is therefore not essential to set it.

The CPU performs the following processing when I/O allocation settings are made.

1) One of the allocations in the following table can be made for each slot of each base unit.

Allocated Number of Points			
Vacant Slot	Input Module	Output Module	Special Function Modules
0	—	—	—
16	16	16	16
32	32	32	32
48	48	48	48
64	64	64	64

2) For slots for which I/O allocations have been made using GPPQ, the I/O allocations set with GPPQ take priority regardless of the modules loaded.

- If a number of points that is fewer than the number of points of the loaded I/O module is set, the actually used number of points of the loaded I/O module is reduced.
  For example, if the loaded module is a 32 point input module but I/O allocation is set for a 16-point input module using GPPQ, it will not be possible to use the latter half 16 points of the input module.
- If a number of points that is greater than the number of points of the loaded I/O module is set, the number of points in excess of the number of points actually occupied become dummy points.
- If the slot where an I/O module is mounted is set as a vacant slot, the mounted I/O module will be unusable.

3) Slots for which I/O allocation is not performed using GPPQ are allocated the number of points of the loaded module.

4) Slots for which I/O allocation is not performed using GPPQ are allocated I/O numbers that are consecutive to those of modules for which I/O allocations have been made.

(3) Cautions

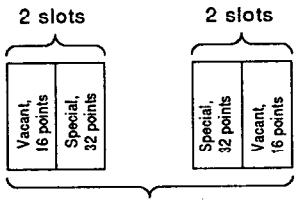
(a) If there is a disparity between the I/O allocations made in the parameter settings and the actually loaded I/O modules, input and output will not be performed normally.

Loaded Module	I/O Allocation	Result
Input	Output	No input
Output	Input	No output
Input/output	Special	CPU error
Special	Input/output	CPU error

(b) The I/O allocation for a slot in which a special function module is mounted must be the correct number of points for that module.

A setting that is incorrect for the module loaded will cause an error.

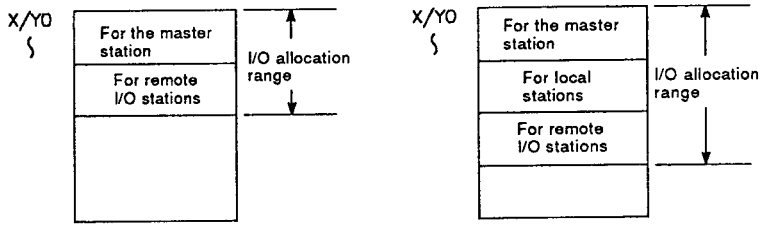
1) A11VC ..... Special: 16 points
2) AI61 ..... Special: 32 points
3) AG62..... Input : Set number of points
4) Modules that occupy 2 slots ... Allocate "vacant, 16 points" and "special, 32 points".



Refer to the user's manual for the special function module used.

(c) When operating a MELSECNET data link, make I/O allocations as follows.

1) At a master station, I/O allocations must be made for the master station and all remote I/O stations.



2) At a local station, make I/O allocations for the local station only.
3) Make the I/O allocation for an input/output combination module (such as A42XY) as if it were an output module.

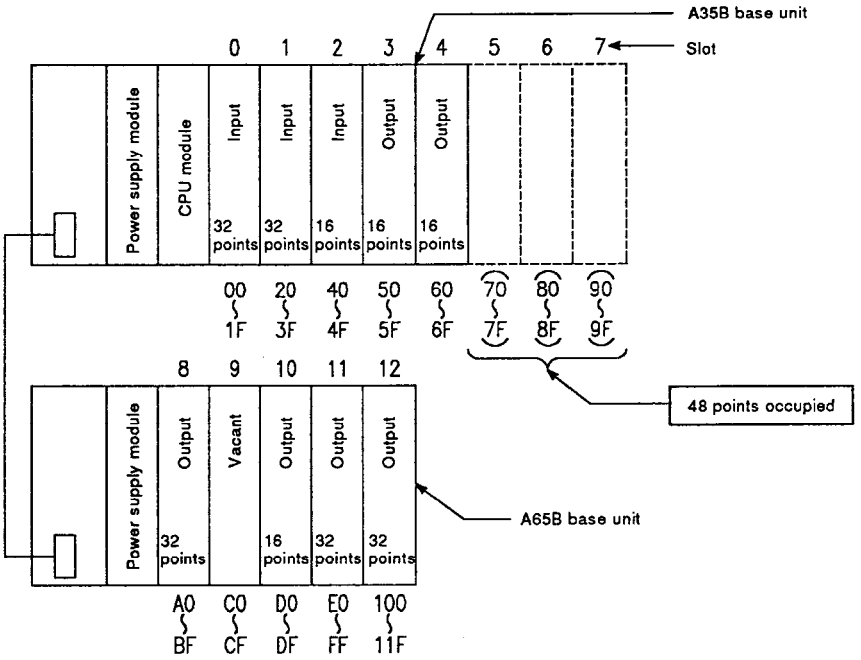
5.4 Example of I/O Number Allocation

An example of the allocation of I/O numbers when a building block type CPU is loaded, and the I/O numbers when I/O allocation is performed using GPPQ is presented here.

(1) Making the 16 points allocated to a vacant slot 0 point or 32 points

When an A35B is used, there are three vacant slots, and you may want to set the allocations for these to 0 point in order to increase the number of I/O points that can be used by the CPU.
Alternatively, you may want to reserve 32 I/O points at a current vacant slot at which a 32 point input/output module is to be installed later, so that the I/O number allocations do not have to be changed.
In order to achieve these requirements, make I/O allocations as follows.

(a) Loading status and I/O numbers



(b) I/O numbers when I/O allocations are made using GPPQ

1) I/O allocation example

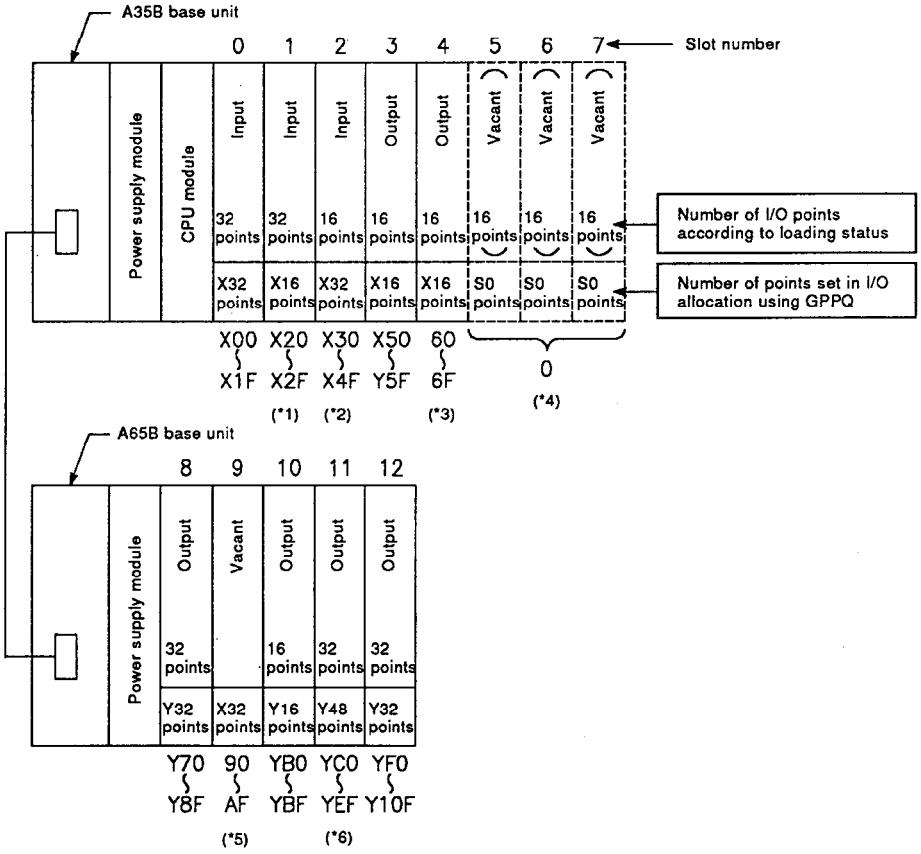
[I/O Allocation] Label :

Slot	Type	Items	istXY	Type Name	
0(0-0)	<Inp >	<32Pt>	[0]	[AX41]	Basic
1(0-1)	<Inp >	<16Pt>	[20]	[AX40]	[A35B]
2(0-2)	<Inp >	<32Pt>	[30]	[AX41]	Power Supply
3(0-3)	<Out >	<16Pt>	[50]	[AY40]	[A61P]
4(0-4)	<Out >	<16Pt>	[60]	[AY40]	Extension Cable
5(0-5)	<Free>	< 0Pt>	[ ]	[ ]	[AC10B]
6(0-6)	<Free>	< 0Pt>	[ ]	[ ]	
7(0-7)	<Free>	< 0Pt>	[ ]	[ ]	
8(1-0)	<Out >	<32Pt>	[70]	[AY41]	Extension 1
9(1-1)	<Free>	<32Pt>	[90]	[ ]	[A65B]
10(1-2)	<Out >	<16Pt>	[B0]	[AY40]	Power Supply
11(1-3)	<Out >	<48Pt>	[C0]	[ ]	[A61P]
12(1-4)	<Out >	<32Pt>	[F0]	[AY41]	Extension Cable
13(1-5)	< >	< >	[ ]	[ ]	[ ]
14(1-6)	< >	< >	[ ]	[ ]	
15(1-7)	< >	< >	[ ]	[ ]	

PgUp:Prev PgDn:Next Esc:Close

Example of I/O Allocations Using GPPQ

2) I/O numbers after I/O allocation using GPPQ



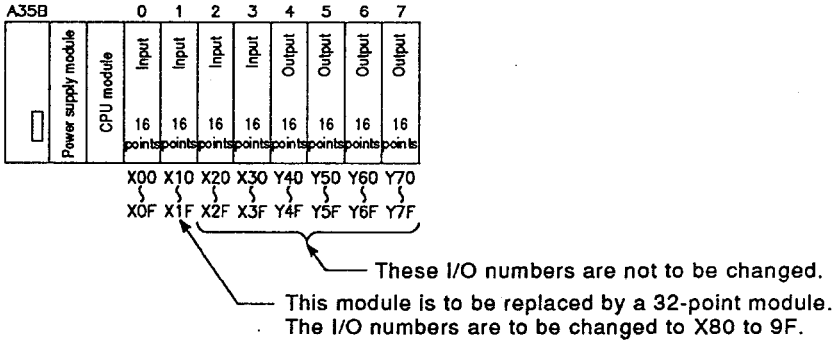
REMARKS

*1: Since 16 points are set, the latter half 16 points of inputs cannot be used.
*2: Since 32 points are set, the points from 40 to 4F are dummy points.
*3: Since "vacant (S), 16 points" is set, the points cannot be used for outputs.
*4: Since "vacant (S), 0 points" is set, three slots are not lost from the number of I/O points.
*5: Since "input (S), 32 points" is set, there are 32 input points.
*6: Since 48 points are set, E0 to EF become dummy points.

(2) Replacing a 16-point input module with a 32-point input module

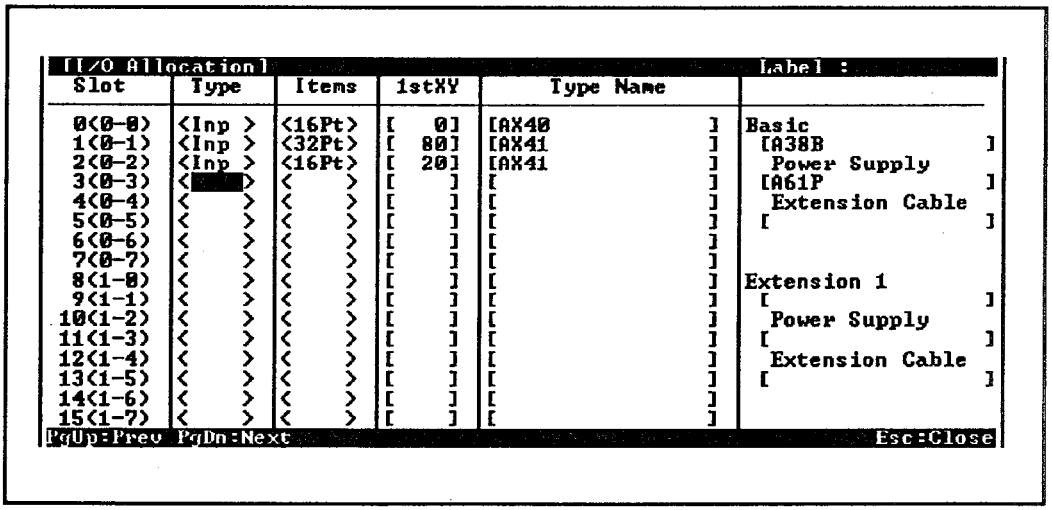
In a system in which a 16-point input module is set, you may want to replace the 16-point input module with a 32-point input module without changing the I/O number allocations as a whole. To do this, make the I/O allocations as follows.

(a) Loading status and I/O numbers before the change



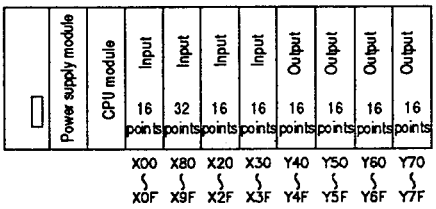
(b) I/O numbers when I/O allocations are made using GPPQ

1) I/O allocation example



Example of I/O Allocations Using GPPQ

2) I/O numbers after making I/O allocations using GPPQ and replacing the module



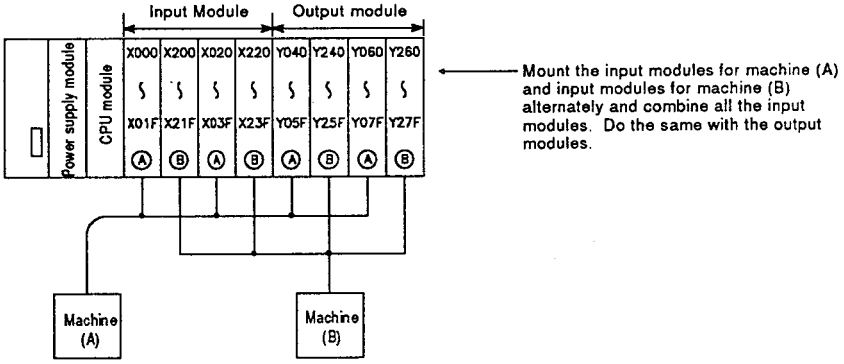
POINT

When the I/O number set for "1stXY" in I/O allocation is changed, also set the "1stXY" for the next module to avoid changing the I/O numbers from the module following the module for which the change was made and subsequent modules.
In the example above, since "20" is set for the "1stXY" for the second slot, consecutive I/O numbers starting from X30 are set for slot 3 onward.

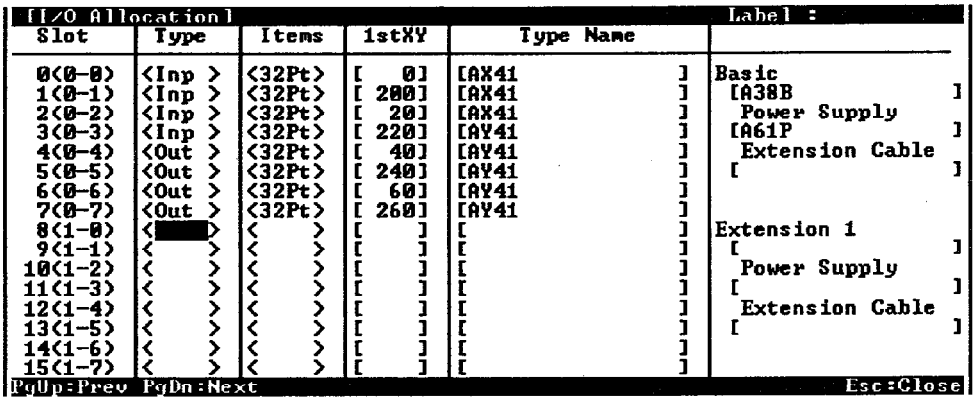
(3) Combining an input module and output module with non-consecutive I/O numbers on a base unit

Assume that, in the situation shown below, where both machine (A) (I/O numbers X0 to X3F, Y40 to 7F) and machine (B) (I/O numbers X200 to X23F and Y240 to 27F) are controlled by the same programmable controller, it is desired to combine input modules and output modules on the base unit. To do this, make the I/O allocations as follows.

(a) Loading status and I/O numbers to be set



(b) Example of I/O allocations using GPPQ



6. DATA COMMUNICATION WITH SPECIAL FUNCTION MODULES

This chapter describes the methods for reading data from a special function module, and writing data to a special function module, with a QnACPU.
Special function modules are modules that allow analog quantities, high-speed pulses, etc., which cannot be processed with input/output modules alone, to be handled by the QnACPU.
To take the example of analog quantities, these are converted to digital values by an analog to digital converter module (which is a special function module) so that they can be used by the QnACPU.
Special function modules have buffer memories in which data input from external sources and data to be output to external destinations are stored.
There are two methods by which data can be read from/written to a special function module with a QnACPU:

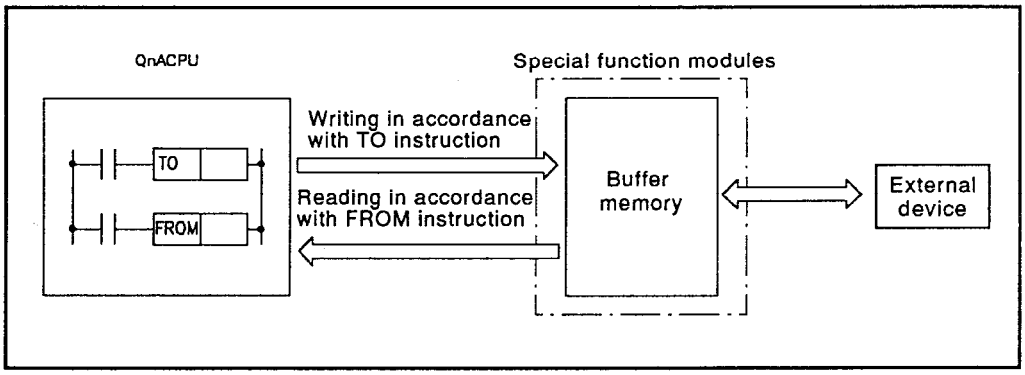
(1) By using FROM/TO instructions

(2) By using special direct devices

These methods are explained in the following sections.

6.1 Reading/Writing Data from a QnACPU Using FROM/TO Instructions

When a FROM/TO instruction is executed, data stored in the buffer memory of a special function module is read, or data is written to the buffer memory of a special function module.



Data Communication with a Special Function Module

When a FROM instruction is executed, data read from the buffer memory is stored in the designated device. When a TO instruction is executed, the data in a designated device is written to the buffer memory.

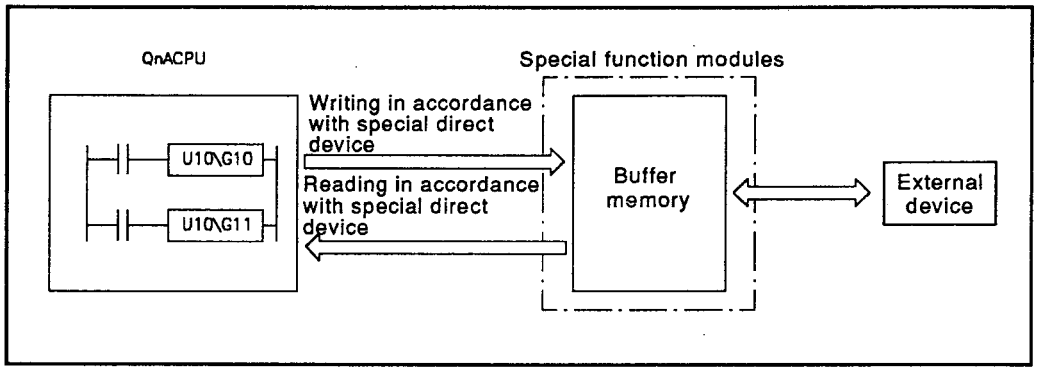
REMARKS

(1) For details on FROM/TO instructions, refer to the QnACPU Programming Manual (Common Instructions).

(2) For details on the buffer memories of special function modules, refer to the manuals for the special function module being used.

6.2 Reading/Writing Data from a QnACPU Using Special Direct Devices

Like FROM/TO instructions, special direct devices are used to read data stored in the buffer memories of special function modules and write data to the buffer memories of special function modules.



Special direct devices are special function module buffer memories expressed as QnACPU devices.

Example: U10\G10

U10 → Indicates the first output number 100 of the special function module. (Hexadecimal)

G10 → Indicates address 10 of the buffer memory. (Decimal)

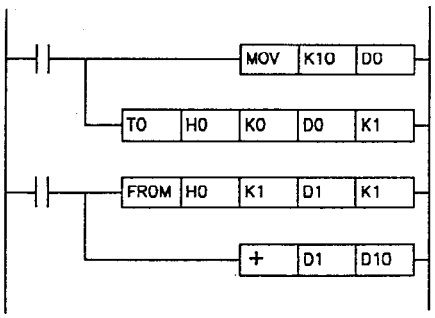
REMARK

For details on special direct devices, refer to the QnACPU Programming Manual (Fundamentals).

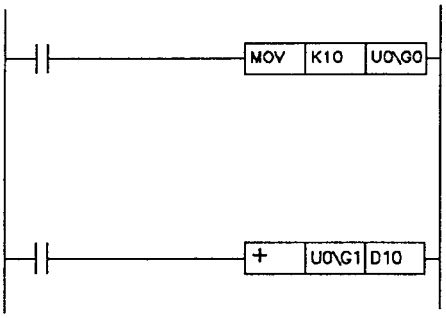
Special direct devices differ from FROM/TO instructions in that the CPU uses the special function module buffer memory directly as a device. This enables a reduction in the total number of program steps. However, the instruction processing speed is the same as for FROM/TO instructions.

Example: Writing data to address 0 of the buffer memory of the special function module mounted at X/Y0, and reading the data of address 1.

(a) With FROM/TO instructions



(b) With special direct devices

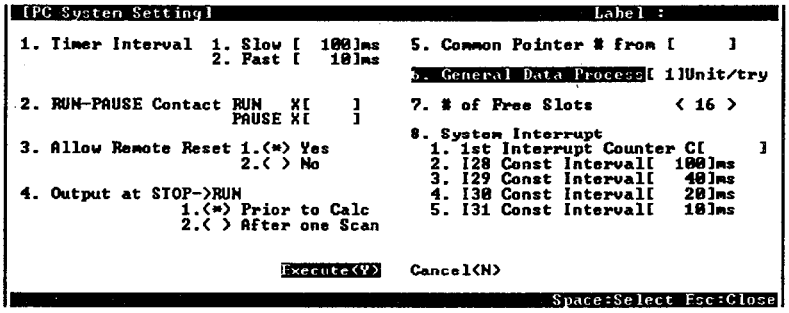


POINT

If special function module data is read frequently in a program, it is better to perform the reading at one location in the program using a FROM instruction and store the read data in a data register, than to use a special direct register for each instruction.
This is because if special direct devices are used the program scan time is extended by the processing to access the special function module at each instruction.

6.3 Processing for Data Communication Requests from Special Function Modules

When a data communication request is received from a special function module such as a serial communication module, the QnACPU executes the processing for the data communication request within the END processing. The QnACPU can process all the data communication requests received in one scan within one END processing, depending on the parameter settings. In this case, the data lag to each module is eliminated, and END processing is lengthened only by the data communication request processing.
Batch processing of data communication requests is set with "6. General Data Process" on the "PC System Setting" screen in the GPPQ parameter mode.
The setting range is 1 to 6 modules, in module units.



7. MELSECNET/MINI(S3) AUTOMATIC REFRESH SETTING

Automatic communication with the batch refresh send/receive data buffer memory area of an AJ71PT32-S3 master module (hereafter called the "master module") is made possible by setting the MELSECNET/MINI(S3) link information, I/O storage devices, etc., in the parameters.
These settings are made in MELSECNET/MINI setting in the GPP parameter mode.

It is possible to use the I/O devices allocated for transmission or reception in MELSECNET/MINI setting without alteration when creating a program. (FROM/TO instructions are not necessary).

POINTS

(1) A maximum of 8 master modules can be set in automatic refresh parameter setting, so up to 8 modules can be handled in automatic refresh processing.
If there are 9 or more modules, the processing for the ninth and later modules has to be performed using FROM/TO instructions.

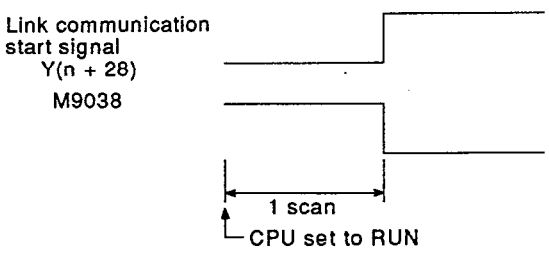
(2) Automatic refresh is not possible for send/receive data for partial refresh I/O modules or the send/receive data of remote terminal modules No.1 through No.14: use FROM/TO instructions for the processing for these.
However, partial automatic refresh is possible with the remote terminal modules listed below.

- Type AJ35PTF-R2 RS-232C interface module
- Type AJ35PT-OPB-M1-S3 mount type operation box
- Type AJ35PT-OPB-P1-S3 portable type operation box

(3) Since the QnACPU automatically turns ON the link communication start signal Y(n+18)/Y(n+28) for the master modules for which the automatic refresh has been set, there is no need to turn this signal on in the sequence program.

(4) Automatic refresh of I/O data is performed in a batch after the QnACPU END instruction has been executed.
(Automatic refresh is executed when the CPU is in the RUN, PAUSE, or STEP RUN state.)

(5) Depending on the connected remote terminal module, the master module may perform processing while the link communication start signal Y(n+18)/Y(n+28) is OFF.
For example, if a type AJ35PTF-R2 RS-232C interface module is used in the no-protocol mode, parameters have to be written to the parameter area (buffer memory addresses 860 to 929) while the link communication signal is OFF.
Since the link communication signal comes ON one scan after the CPU is set to the RUN state, write the parameters in the first scan.



(6) If the hardware error signal X(n+0)/X(n+20) or ROM error signal X(n+8)/X(n+28) of a master module for which automatic refresh has been set comes ON, the QnACPU does not perform automatic refresh processing.

(7) When making the settings, ensure that there is no duplication between receive data refresh devices and send data refresh devices.

(1) The parameter settings to be made for automatic refresh, the setting ranges, setting descriptions, and buffer memory address of the master module communicating data with the AnUCPU are tabled below.
Set parameters for all of the number of master modules used.

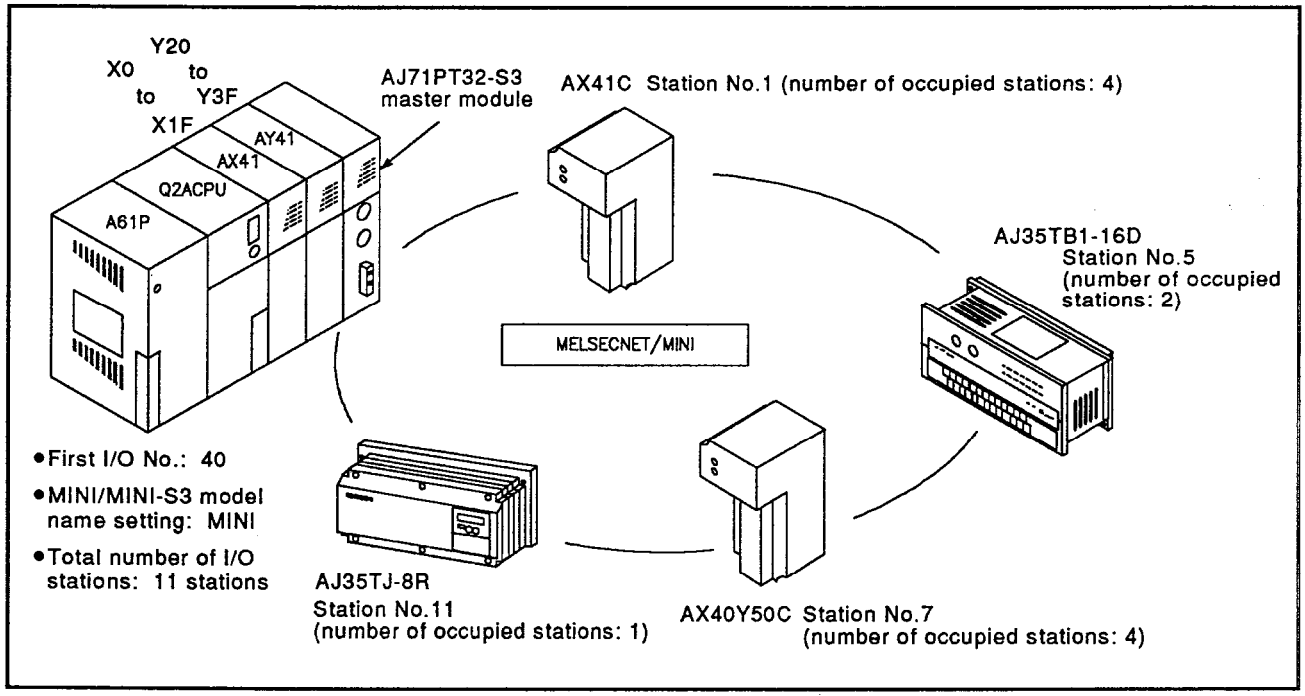
Master Module I/O Signal	Master Module Buffer Memory Address	Item	Setting Range	Description	Default Value
—	—	Number of master modules	0, 1 to 8	• Set the total number of master modules used. Set "0" if automatic refresh is not to be used.	Follow the settings made in I/O allocation in the parameter mode.*3
—	—	First I/O No.	Number of CPU I/O points	• Set the first I/O number of those where the master module is mounted.	
—	—	MINI/MINI-S3 model name setting	• MINI or MINI-S3	• MINI ..... In I/O dedicated mode (32 points occupied) • MINI-S3 ..... In extension mode (48 points occupied)	
—	0	Total number of remote I/O stations	0 to 64 stations	• Set only when "MINI" is set. • When "MINI-S3" is set, no setting is required since the number of initial ROMs of the master modules is effective (any setting made is ignored).	
—	110 to 141	Receive data storage devices	• X • M, L, B, T, ST, C, D, W, R, ZR, none (bit devices designated in multiples of 16)	• Set the devices used to store receive/send data for batch refresh. • Set by designating the first device number. • An automatic refresh area for the applicable number of stations, starting from the first device number, is allocated. (8 points/station x 64 stations = 512 points ... bit devices) *2 • You are recommended to use the X/Y remote I/O range for devices.	X1000 to X11FF
—	10 to 41	Send data storage devices	• Y • M, L, B, T, ST, C, D, W, R, ZR, none (bit devices designated in multiples of 16)		Y1000 to Y11FF
—	1	Retry count	0 to 32 retries	• Set the retry count when a communication error occurs. • If communication is restored within the retry count, no error is output.	5 retries
*1 Y(n+1A)	—	FROM/TO response designation	Link priority, CPU priority (Priority selection for access to the buffer memory of the master module)	(1) Link priority ....MINI-S3 link access is given priority. FROM/TO instructions are put on standby during link access. • Receive data refreshed in accordance with the same timing can be read. • A maximum FROM/TO instruction waiting time of (0.3 ms + 0.2 ms x number of partial refresh stations) arises. (2) CPU priority ...CPU FROM/TO instructions are given access priority. They can access the buffer memory by interruption during link access. • Depending on the timing, receive data may be read part way through I/O refresh processing. • There is no waiting time for FROM/TO instructions.	CPU priority
*1 Y(n+1B)	—	Faulty station data clear designation	Latch, clear (receive data)	• Latch ..... The receive data for batch refresh/partial refresh is latched. • Clear ..... All points are turned OFF.	Clear
—	100 to 103 195	Faulty station detection	M, L, B, T, ST, C, D, W, R, ZR, none (bit devices designated in multiples of 16)	• Set the first device of those to store the faulty station detection data. • MINI ..... 4-word allocation MINI-S3 ..... 5-word allocation	No setting
—	107 196 to 209	Error No.	T, ST, C, D, W, R, ZR	• Set the first device of those to store the error code when an error occurs. • MINI ..... 1-word allocation MINI-S3 ..... (1 + number of remote terminals)-word allocation	No setting

Master Module I/O Signal	Master Module Buffer Memory Address	Item	Setting Range	Description	Default Value
—	4	Line error check setting (Line error)	• Send test message (test) • Send OFF data (OFF) • Send data immediately prior to line error (latch)	• Set the data sending method for checking the location of the error when a line error occurs.	Latch
—	—	Operation at CPU STOP	Stop, continue	• Set the operating status when the CPU is in the STOP state.	Stop

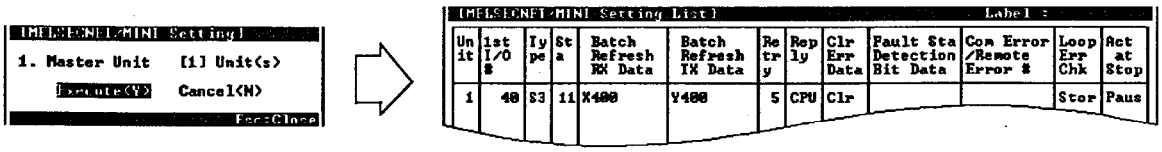
*1: "n" is determined by the mounting position of the master module.
*2: If the total number of I/O stations is an odd number, an extra allocation equivalent to one station is made for each storage device.
*3: If the setting area for the number of master modules is left blank, automatic refresh can be performed without making this setting. However, model name registration is required in I/O allocation.

(2) The setting of send/receive data storage devices is explained here by reference to the system example shown below.

Example: Using devices from X/Y400 onward for remote I/O stations

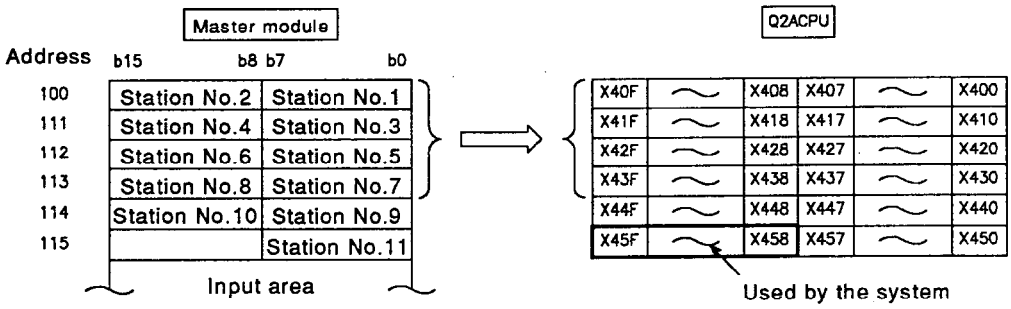


An example of the GPP function software parameter settings for the system above is shown below.



The send/receive data storage devices in the system example are as follows.

(a) Receive data storage devices



(1) Set the b0 device number of station No.1 (X400) as the send data storage device.

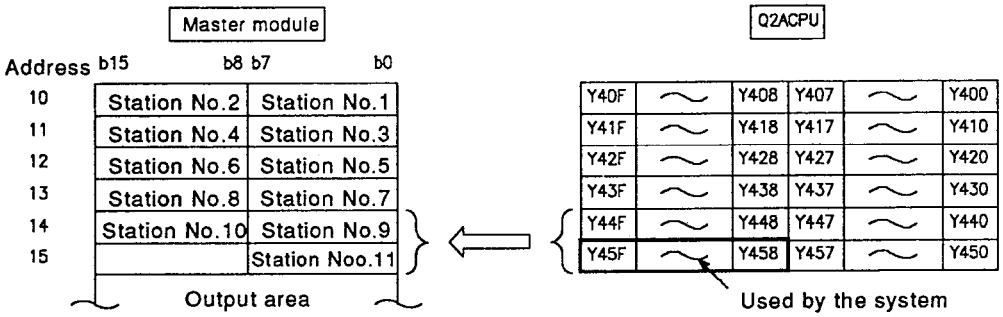
(2) The occupied receive data storage devices are X400 to X45F.
Since the total number of stations is an odd number, one extra station is used.

(3) The device numbers of the connected I/O module are as follows:

Station numbers 1 to 4	AX41C ⟶ X400 to X41F
Station numbers 5 and 6	AJ35TB-16D ⟶ X420 to X42F
Station numbers 7 and 8	AX40Y50C ⟶ X430 to X43F

X440 to X45F are also refreshed at the same time and are always OFF.
Do not use X440 to X45F in the sequence program.

(b) Send data storage devices



(1) Set the device number corresponding to b0 of station No.1 (Y400) as the send data storage device.

(2) The occupied send data storage devices are Y400 to Y45F.
Since the total number of stations is an odd number, one extra station is used.

(3) The device numbers of the connected output modules are as follows.

Stations numbers 9 and 10	AX40Y50C ⟶ Y440 to Y44F
Station number 11	AJ35TJ-8R ⟶ Y450 to Y457

Y400 to Y43F and Y458 to Y45F are also refreshed at the same time.
However, no outputs are made.

POINTS

(1) If the same device type is used for the send data storage devices and receive data storage devices, make sure that there is no duplication of device numbers.
If B0 is set as the receive data storage device in the example system, B0 to B5 is occupied as the device range.
Set B60 or higher as the send data storage device.
If B60 is set as the send data storage device, the device range is B60 to BBF.

(2) If a bit device is set as the send/receive data storage device, always set a device number that is a multiple of 16.

Example:
X0, X10, ... X100, ...
M0, M16, ... M256, ...
B0, B10, ... B100, ...

(3) The used device range is (8 points) x (number of stations).
However, an additional 8 points is required if the number of stations is an odd number.

8. DEBUGGING FUNCTIONS

8.1 Function List

The QnACPU has a variety of functions that are convenient when debugging.
These debugging functions are listed below.

Item	Description	Refer to
Monitor function	Function for reading CPU programs, device statuses, etc., from a peripheral device capable of GPP functions	Section 8.2
Write during RUN	Function for writing a program while the CPU is in the RUN status	Section 8.3
Execution time measurement	Functions for displaying the processing time of a program being executed	Section 8.4
Program list monitor	Functions for displaying the processing time of a program being executed	Section 8.4.1
Interrupt program list monitor	Function for displaying the number of times an interrupt program is executed	Section 8.4.2
Scan time measurement	Function for measuring the execution time of any section of a program	Section 8.4.3
Sampling trace function	Function for continually collecting the data of devices in accordance with a timing set at the CPU	Section 8.5
Status latch function	Function for collecting device data at the instant the function is designated	Section 8.6
Step run	Functions for running one step or one part of a program, or running a program with a part skipped	Section 8.7
Step execution	Function for running a program step by step	Section 8.7.1
Partial execution	Function for executing a designated part of a program	Section 8.7.2
Skip function	Function for executing a program with a designated part skipped	Section 8.7.3
Program trace function	Function for collecting the program execution status	Section 8.8
Simulation function	Function for simulating execution in isolation from the input/output modules and special function modules	Section 8.9
Debugging by several people	Function for debugging from several peripheral devices capable of GPP functions simultaneously	Section 8.10
Monitoring trace function	Function for collecting device data at a peripheral device capable of GPP functions in accordance with the designated timing	—

For details on the operation for each function, refer to the SW0IVD-GPPQ Operating Manual (Online).

8.2 Monitor Function

This function reads CPU programs and device statuses to a peripheral device capable of GPP functions.

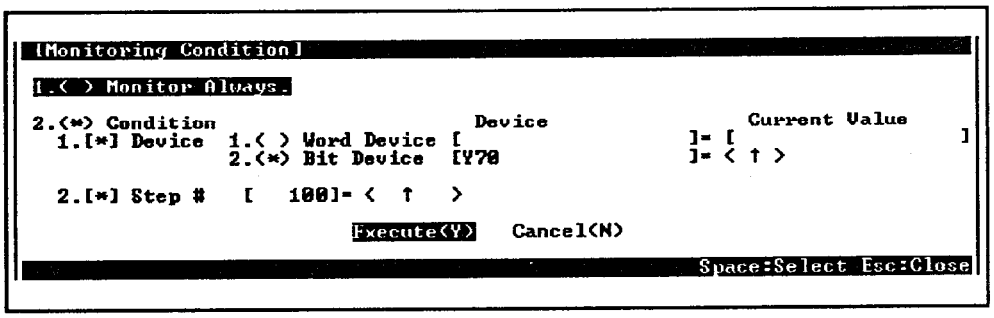
Application

This function is used to set monitoring conditions for monitoring the operating states of the programmable controller in accordance with a precise timing. Apart from execution of monitoring at END processing, it is possible to set the step number to be monitored, the ON/OFF status for that step, and a device status, as the "Monitoring Condition".
By setting a monitor stop condition, it is possible to hold a monitored screen in accordance with a precise timing.

Function Description

(1) Allows setting of the condition under which monitoring is executed.

All operations are performed using the monitor/test menu in the ladder mode. An example of setting a monitor condition is shown below.



The following is an explanation of the screen above:
The monitoring condition can be set as either "1. ( ) Monitor Always." or "2. ( ) Condition".

(a) When "1. ( ) Monitor Always." is set

The collection timing for monitor data is every scan, after END processing at the PC CPU.

(b) When "2. ( ) Condition" is set

"1. [ ] Device" and "2. [ ] Step #" can be set.

1) When "2. [ ] Step #" only is set
The monitor data collection timing is such that data is collected only when the status immediately before execution of the designated step is the designated status.
The possible designations for execution status are as follows.
(I) Executed on switching from OFF to ON status: < ↑ >
(II) Executed on switching from ON to OFF status: < ↓ >
(III) Executed continually while ON : <ON>
(IV) Executed continually while OFF : <OFF>
(V) Executed continually regardless of status : <Always>

REMARKS

(1) When "Step # [0]" is designated, the execution condition must be set as "Always".

(2) When "2. [ ] Step #" is not designated, the collection timing for monitor data is every scan, after END processing by the PC CPU.

2) When only "1. [ ] Device" is set
Either "1. ( ) Word Device" or "2. ( ) Bit Device" can be designated.

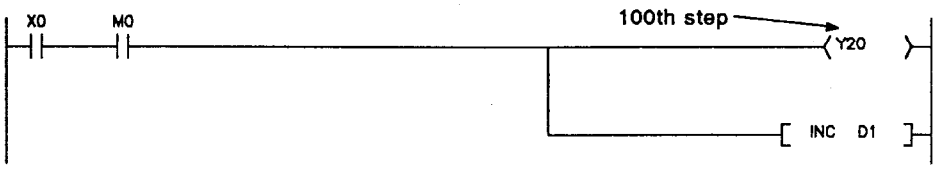
(a) When "1. ( ) Word Device" is designated
The collection timing for monitor data is such that monitor data is collected when the present value of the designated word device attains the designated value.
The method for designating the present value is as follows.
(I) Decimal designation: [K Decimal number]
(II) Hexadecimal designation: [H Hexadecimal number]

(b) When "2. ( ) Bit Device" is designated
The collection timing for monitor data is such that data is collected when the designated bit device attains the designated execution status.
The possible designations for execution status are as follows.
(I) At leading edge: < ↑ >
(II) At trailing edge : < ↓ >

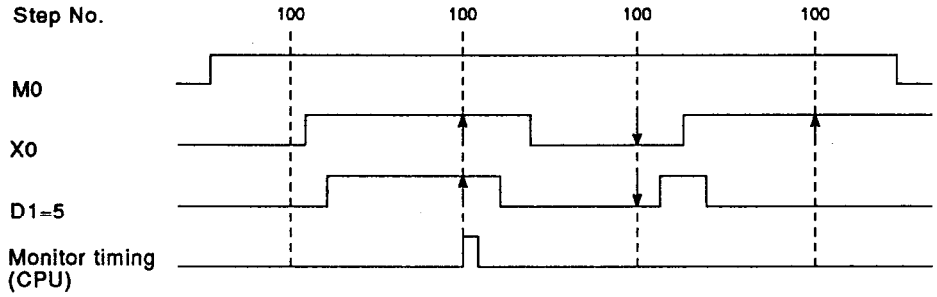
3) When "2. Step # [ ]" and "1. [ ] Device" are designated.
The collection timing for monitor data is such that data is collected when the status immediately before execution of the designated step or the execution status (present value) of the designated bit device (word device) attains the designated status.

POINTS

(a) In the ladder block shown below, assuming that the detailed condition is set as follows: "Step No. [100] = < ↑ > , Word device [D1] = [K5]".



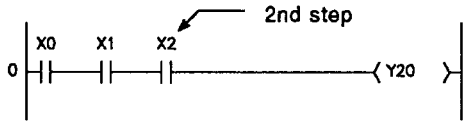
The monitor timing is shown below.
However, the monitoring interval at a peripheral device capable of GPP functions depends on the processing speed of that peripheral device. Even if data changes occur faster than the monitor interval, data can be collected only once during the interval.



(b) Assume that "Step No. [2] = < ON >" is designated as the detailed condition in the case of the ladders shown below:

Ladder mode

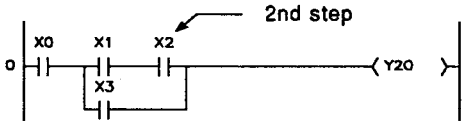
(1)



List mode

0	LD	X0
1	AND	X1
2	AND	X2
3	OUT	Y20

(2)

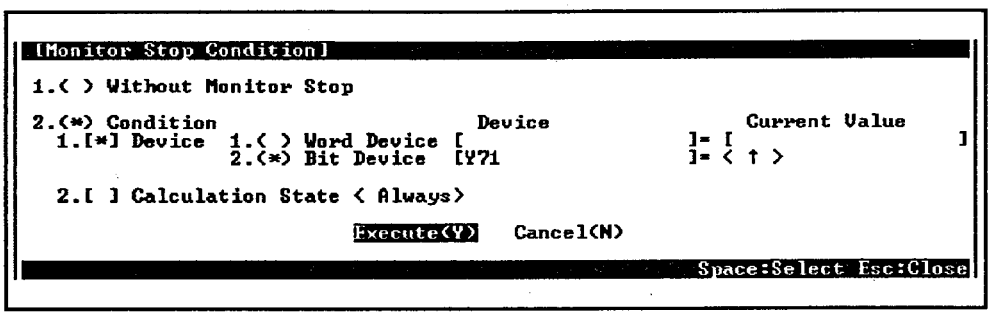


0	LD	X0
1	LD	X1
2	AND	X2
3	OR	X3
4	ANB	
5	OUT	Y20

In this case the monitor execution differs for the two ladders. For (1) it is "X0 and X1 both ON" and for (2) it is "X1 ON (regardless of ON/OFF status of X0)".
If a step part way through an AND/OR block is designated for a monitor condition, the monitor data collection timing is such that data is collected when the status immediately before execution of the step designated from the LD instruction in the block becomes the designated status.

(2) A monitor stop condition can be set.

All operations are performed on the monitor/test screen in the ladder mode.
An example of the setting for a monitor stop condition is shown below.



The following is an explanation of the screen above:
Either "1. ( ) Without Monitor Stop" or "2. ( ) Condition" can be set for the monitor stop condition.

(a) When "1. ( ) Without Monitor Stop " is set

Monitoring is stopped when [Esc] key is pressed.

(b) When "2. ( ) Condition" is set

"1. [ ] Device" and "2. [ ] Calculation State" can be set.

1) When "2. [ ] Calculation State" is set
The monitor stop timing is such that monitoring stops when the execution condition of the step designated for the monitor condition attains the designated status.
The possible designations for execution status are as follows.
(I) Executed on switching from OFF to ON status : < ↑ >
(II) Executed on switching from ON to OFF status : < ↓ >
(III) Executed continually while ON : <ON>
(IV) Executed continually while OFF : <OFF>
(V) Executed continually regardless of status : <Always>

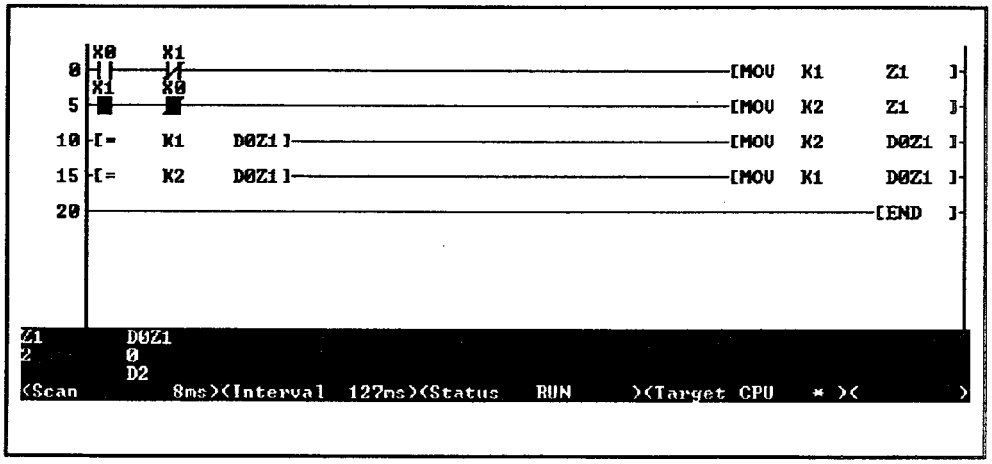
If "2. [ ] Calculation State" isn't set, the timing for monitor stop is such that monitoring is stopped after PC CPU END processing.

2) When "1. [ ] Device" is set
Either "1. ( ) Word Device" or "2. ( ) Bit Device" can be set.

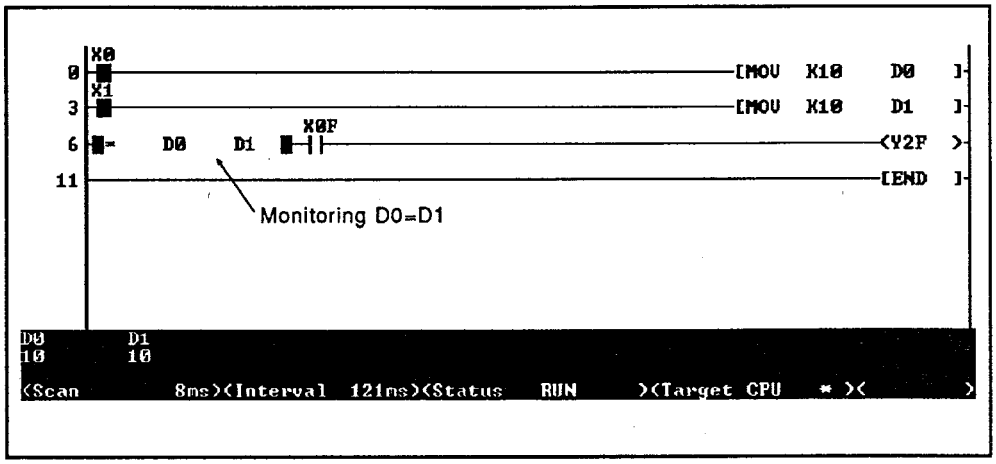
(a) When "1. ( ) Word Device" is set
The monitor stop timing is such that monitoring stops when the present value of the designated word device attains the designated value.
The method for designating the present value is as follows.
(I) For decimal (word) designation : [K Decimal number]
(II) For hexadecimal (word) designation : [H Hexadecimal number]
(III) For decimal (double word) designation : [K Decimal number Space L]
(IV) For hexadecimal (double word) designation : [H Hexadecimal number Space L]
(V) For real number designation: [E Real number]

(b) When "2. ( ) Bit Device" is designated
The monitor stop timing is such that monitoring stops when the execution status of the designated bit device becomes the designated status.
The possible designations for execution status are as follows.
(I) At leading edge: < ↑ >
(II) At trailing edge : < ↓ >

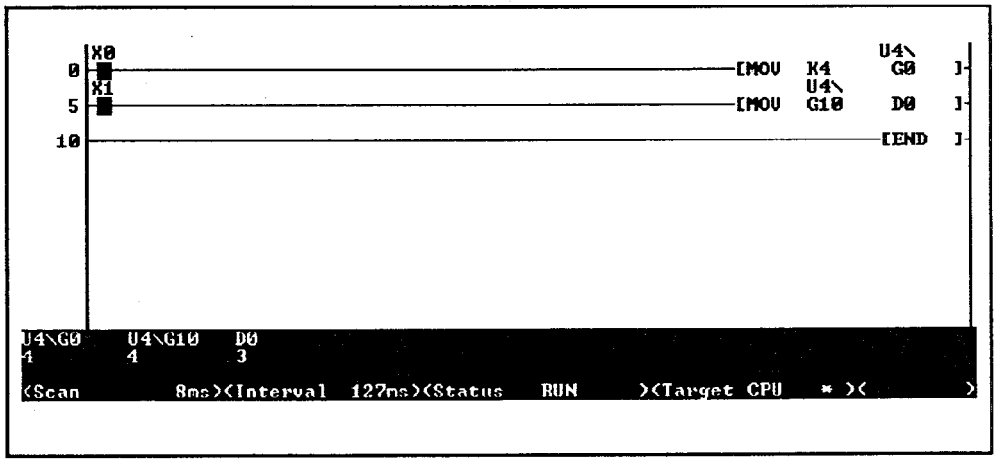
(3) In the case of devices for which index qualifications have been made, the index qualified value is monitored.
An example of this type of monitoring is shown below.



(4) The ON status of comparison instructions can be monitored.
An example of this type of monitoring is shown below.



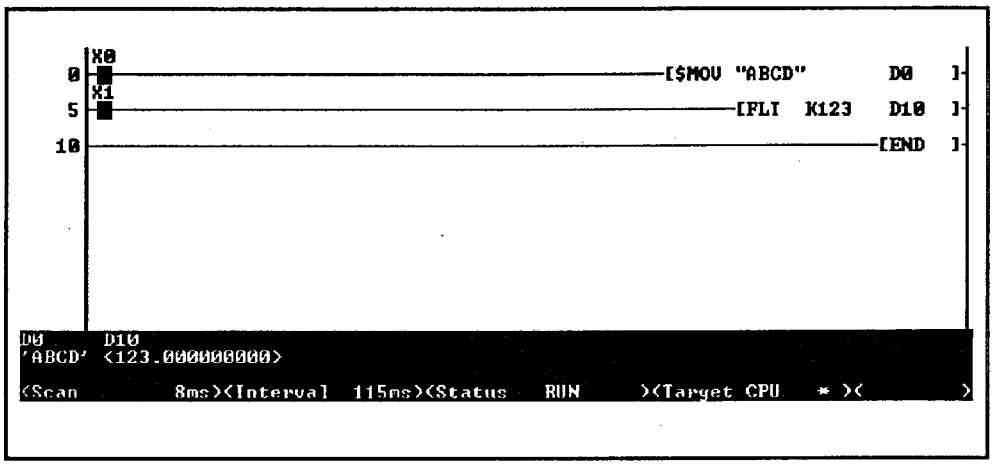
(5) The devices of special function modules can be monitored.
An example of this type of monitoring is shown below.



REMARK

To monitor devices of special function modules, set "2. Buffer Memory 1. Monitor" for "5/Monitor Target Setting" under the ladder mode "Option" menu.

(6) Real numbers and character strings can be monitored.
An example of this type of monitoring is shown below.



(7) The devices that can be monitored are the following.

(a) Bit devices : X, FX, DX, Y, FY, DY, M, L, F, SM, V, B, SB, T (contact), T (coil), ST (contact), ST (coil), C (contact), C (coil), J□\X, J□\Y, J□\B, J□\SB, BL□\S

(b) Word devices: T (present value), ST (present value), C (present value), D, SD, FD, W, SW, R, Z, ZR, U□\G, J□\W, J□\SW

(8) The following qualifications are possible with respect to the devices listed above.

(a) Bit devices : X, FX, Y, FY, M, L, F, SM, V, B, SB, T (contact), ST (contact), C (contact), J□\X, J□\Y, J□\B, J□\SB, BL□\S

(b) Word devices: T (present value), ST (present value), C (present value), D, SD, FD, W, SW, R, Z, ZR, U□\G, J□\W, J□\SW

The following qualifications are possible with respect to the devices listed above.

- Digit designation for bit devices
- Bit number designation for word devices

Cautions

(1) When a monitoring is performed with a monitor condition set, the file displayed at the device running GPPQ is monitored.
Make sure that the file name used with GPPQ is the same as the file name when monitoring is performed by executing "Newly from PC".

(2) When the buffer memory of a peripheral device is read by designating a direct device, $FFFF_H$ is monitored if the peripheral device is faulty or not connected.

(3) When monitoring file registers, $FFFF_H$ is monitored if no file register designation is made.

(4) Before monitoring, make sure that the device allocations of the CPU and GPPQ agree.

(5) When monitoring the local devices in program files, detailed conditions (step number and device condition) must be set for each program file.

(6) When monitoring the buffer memory of a special function module, the scan time is prolonged in the same way as it is when a FROM/TO instruction is executed.

(7) Several people can perform monitoring at the same time.
However, the following considerations apply when this is done:

- High-speed monitoring can be made possible by increasing the system area by 1 k steps for each monitor file for other station use when formatting the internal memory.
Monitor files can be set for up to 15 other stations. However, the memory area available for executing programs decreases by the area occupied by these files.
- Only one person can set the detailed conditions for monitoring.

(8) The detailed conditions for monitoring can only be set in ladder monitoring.

(9) If the same device is designated for both a monitor condition and monitor stop condition, also designate the "ON" or "OFF" status.

8.3 Write During RUN

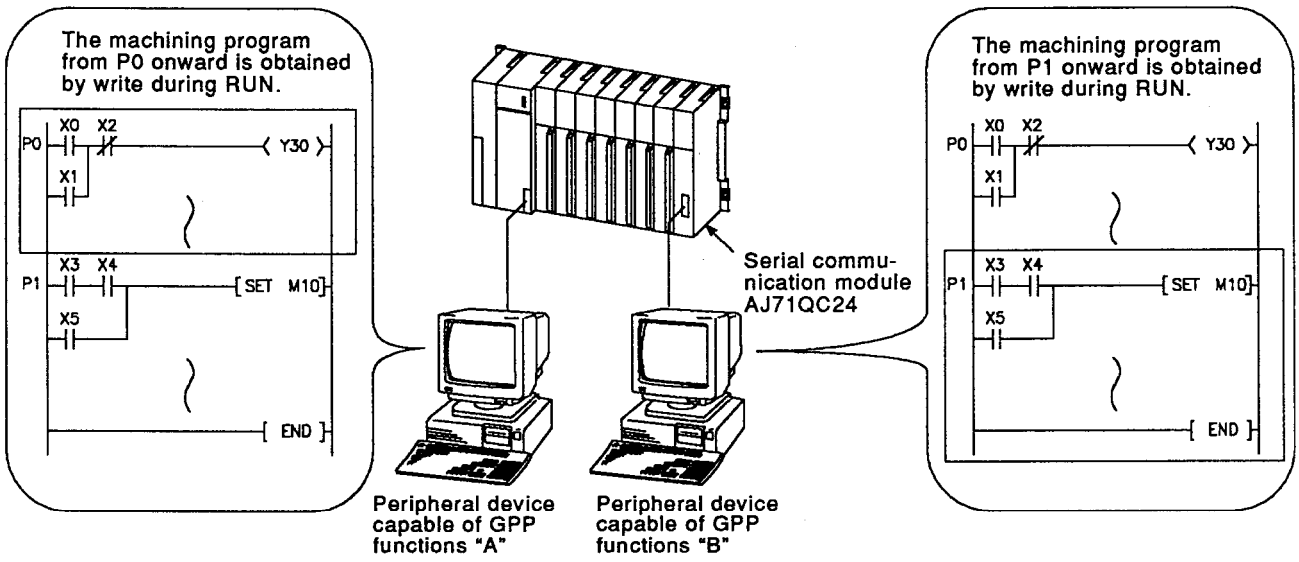
This is a function for writing a program to the CPU while the CPU is in the RUN state.

Application

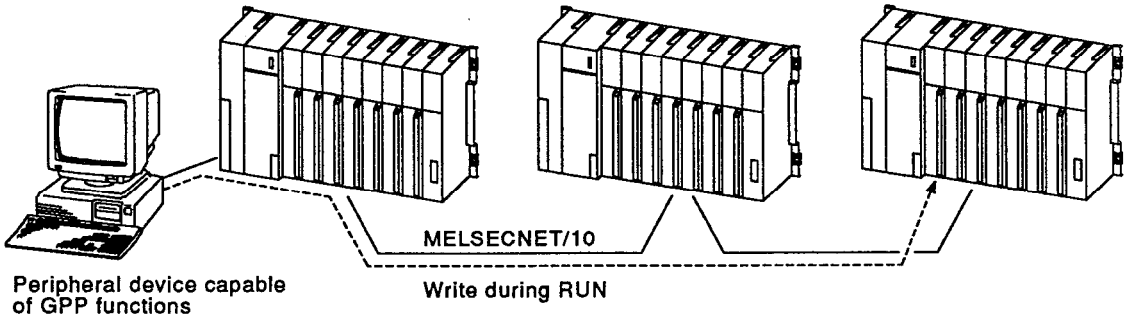
Used to change a program without stopping program execution.

Function Description

(1) Write during RUN is possible from multiple peripheral devices capable of GPP functions with respect to one file.
In order to do this, pointer designations for the programs to be written from the peripheral devices capable of GPP functions using write during RUN must be made in the program in advance.
The example below shows a case where peripheral device capable of GPP functions "A" performs write during RUN from P0, and peripheral device capable of GPP functions "B" performs write during RUN from P1. The program enclosed in the frame [ ] is the program subject to write during RUN.



(2) It is possible to write programs from peripheral devices capable of GPP functions that are connected to other stations in the network during the RUN state.



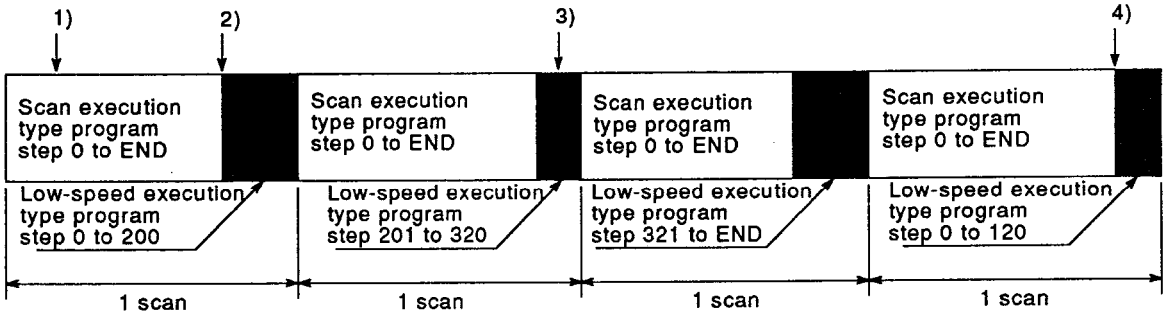
Cautions

The cautions relating to write during RUN are listed below.

(1) The only memory that can be used for write during RUN is the internal memory. If write during RUN is performed during a boot operation, also write the program to the memory card in the STOP state.
If the boot operation is started again without having written the program to the memory card, the program before write during RUN was performed is run.

(2) The maximum number of steps that can be handled in one write during RUN operation is 512.

(3) During low-speed program execution, write during RUN is started when execution of all low-speed programs is completed. Also, execution of low-speed programs is suspended during write during RUN.



1) : Scan execution type program write during RUN command

2) : Execution of scan execution type program write during RUN

3) : Low-speed execution type program write during RUN command

4) : Execution of low-speed execution type program write during RUN

8.4 Execution Time Measurement

This is a function for displaying the processing time of the program being executed.

Application

Used to determine the influence of the processing time of each program on the total scan time when making system adjustments.

Function Description

Execution time measurement includes the following three functions. For explanations of each function, refer to Sections 8.4.1 through 8.4.3.

- Program list monitor
- Interrupt program list monitor
- Scan time measurement

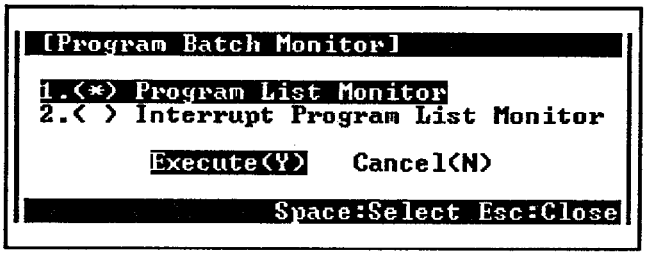
8.4.1 Program list monitor

This is a function for displaying the processing time of the program being executed.

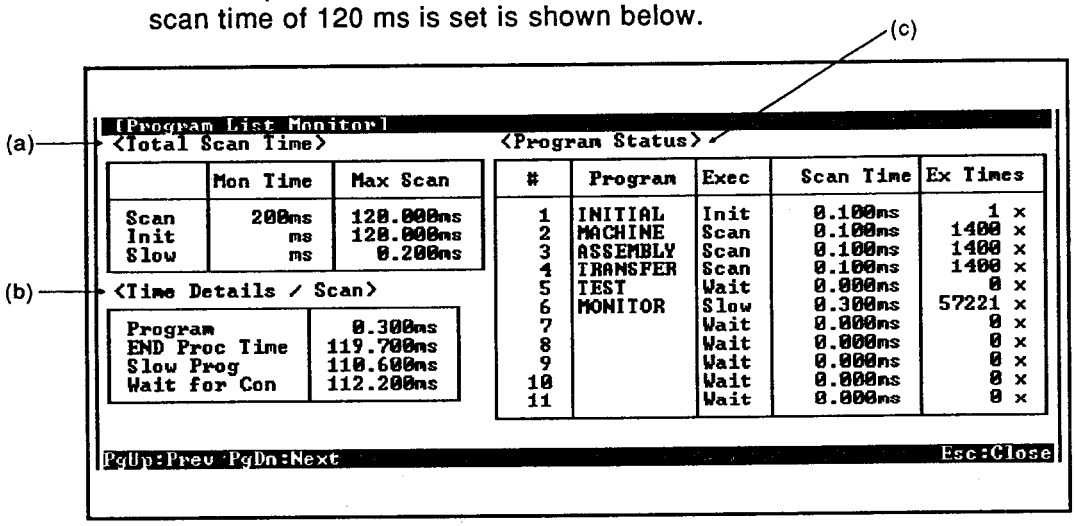
Function Description

The scan time, execution count, and processing time for each item can be displayed for each program.
The operations for this function are performed under the monitor/test menu in the monitor status of the ladder mode.

(1) Select "Program Batch Monitor".



(2) Select "Program List Monitor".
An example of execution of the program list monitor when a constant scan time of 120 ms is set is shown below.



The following is an explanation of the screen above:

(a) "Total Scan Time"

The times set in "5.( ) PC RAS Setting" for monitor time and scan time total are displayed here for each program type.

1) "Mon Time"
The monitor times for scan execution type programs, initial execution type programs, and low-speed execution type programs are displayed here.
If the scan time exceeds the time displayed here, a watchdog timer error occurs at the CPU.

2) "Max Scan"
The total time for the items listed under "Time Details / Scan" is displayed here.

(b) "Time Details / Scan"

The scan time details are displayed here.

1) "Program"
The total execution time of scan execution type programs is displayed here.

2) "END Proc Time"
The END processing time is displayed here.

3) "Slow Prog"
When an execution time for low-speed execution type programs is set, the total execution time for low-speed execution type programs is displayed here.

4) "Wait for Con"
When constant scan is set, the constant scan waiting time is displayed here. However, if an execution time for low-speed execution type programs is also set, the time displayed is 0.000 ms.

(c) "Program Status"

The times set in "9. ( ) Auxilliary Setting" in the parameter mode are displayed here.

1) "Program"
The program names are displayed here in the order they were set in the parameters.

2) "Exec"
The types of the programs set in the parameters are displayed here.

3) "Scan Time"
The actual scan times (present values) are displayed here. In the program stop (standby) status, the scan time is displayed as 0.000 ms.

4) "Ex Times"
The execution count is displayed here, taking the point at which measurement was started to be "0" in the count (on reaching the maximum count of 65536, the count returns to "0"). The execution count is retained even when program execution is stopped.

8.4.2 Interrupt program list monitor

This function displays the execution counts of interrupt programs.

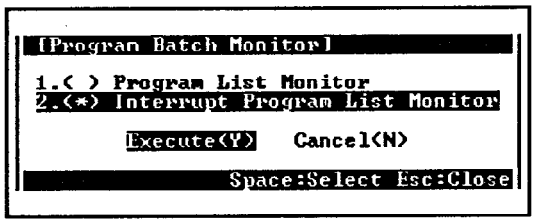
Application

Used to check the execution status of interrupt programs.

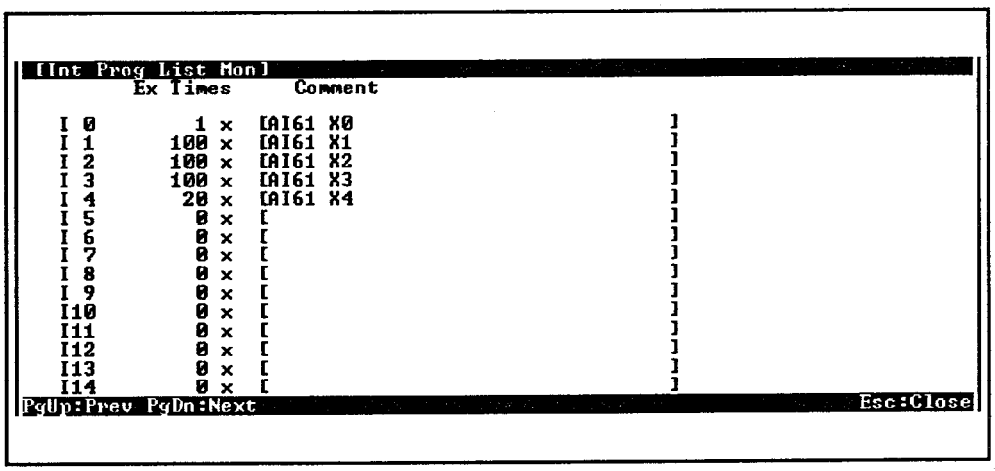
Function Description

Allows display of the execution counts of interrupt programs.
All operations are performed using the monitor/test menu in the ladder mode.

(1) Select "Program Batch Monitor".



(2) Select "Interrupt Program List Monitor".
An example of the display when the interrupt program list monitor function is run is shown below.



The following is an explanation of the screen above:

(a) "Ex Times"

The number of times the program has been executed, taking the point when monitoring started as "0" in the count, is displayed here. (On reaching the maximum count of 65536, the count is returned to "0".)
The count is cleared to zero on switching the operating state to RUN.

(b) "Comment"

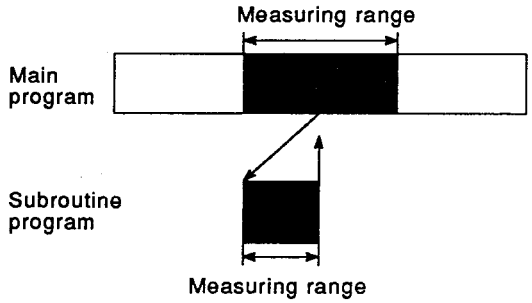
The comments set in the documentation mode are displayed here.

8.4.3 Scan time measurement

This function displays the processing time for any required part of a program.

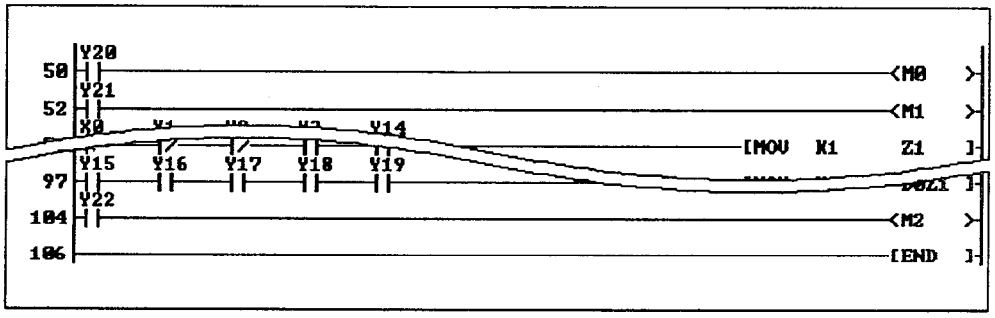
Function Description

Allows measurement of the execution time of any part of the program in a program file.
The function can also be used to measure times within subroutine programs and interrupt programs.
If an interrupt program exists in the monitored section, the processing for that program is included in the measured time.

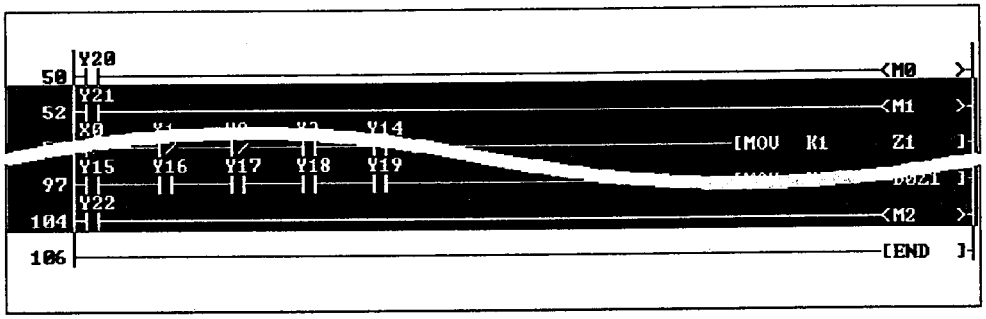


All operations are performed using the monitor/test menu in the ladder mode.

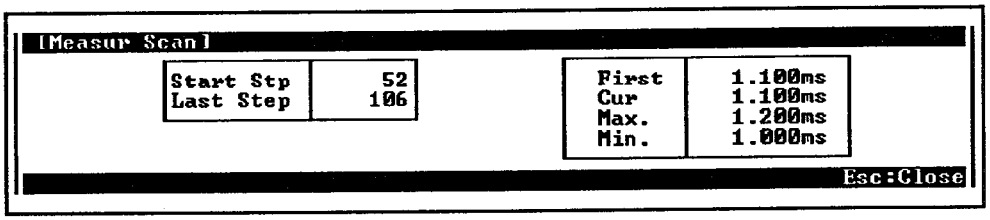
(1) Select "Measur Scan".



(2) Designate the scan time measurement range (the designated part is highlighted).



(3) The scan time measurement results are displayed.



Cautions

(1) Make sure that the start step is lower than the end step in the setting.

(2) Times that span different program files cannot be measured.

(3) If the measured time is less than 0.100 ms, it is displayed as 0.000 ms.

8.5 Sampling Trace Function

This function collects the data of devices continually in a designated timing.

Application

Can be used to check the changes in the contents of the devices used in a program in accordance with a designated timing during debugging.
This shortens debugging time.

Function Description

(1) Function

(a) The sampling trace function samples the contents of a designated device in a constant time interval (the sampling cycle) and stores the trace results in a sampling trace file in a memory card.

(b) The devices that can be traced are listed below.

1) Bit devices : X, FX, DX, Y, FY, DY, M, L, F, SM, V, B, SB, T (contact), T (coil), ST (contact), ST (coil), C (contact), C (coil) , J□\X, J□\Y, J□\B, J□\SB, BL□\S . . . . . Max. 50 points

2) Word devices: T (present value), ST (present value), C (present value), D, SD, FD, W, SW, R, Z, ZR,U□\G, J□\W, J□\SW . . . . . Max. 50 points

(c) The sampling trace file stores the trace condition data and trace execution data required to execute the sampling trace. After a trace has been started at a peripheral device capable of GPP functions, it is continued until the set number of traces is completed.

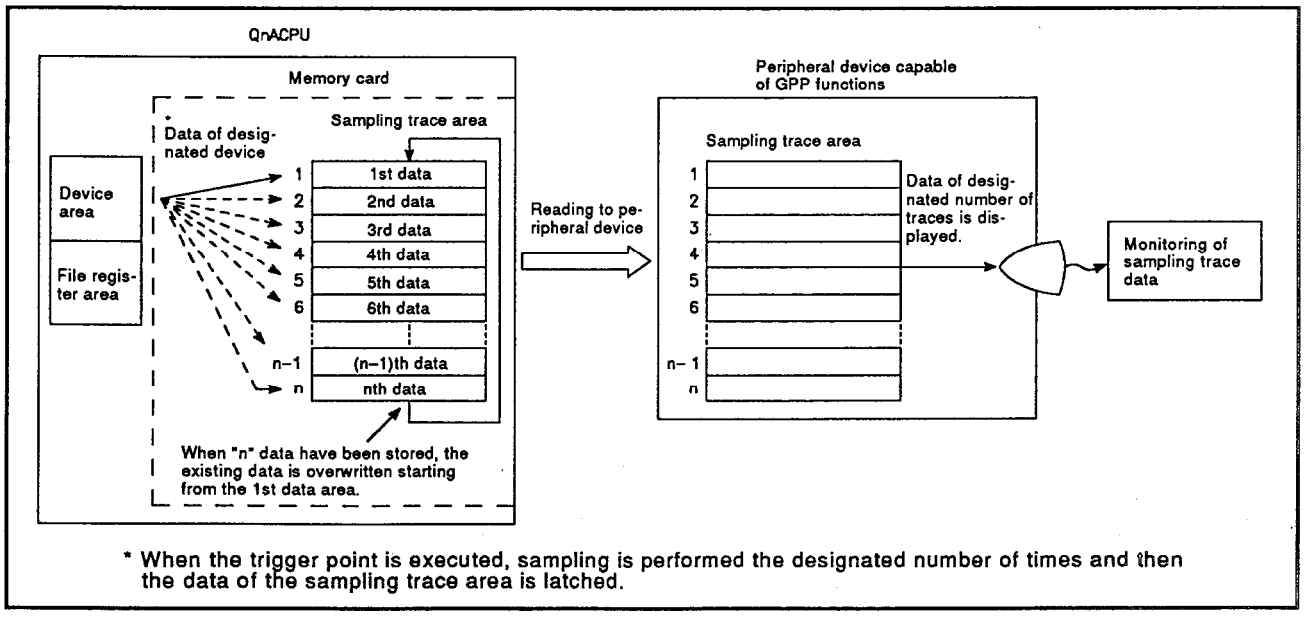
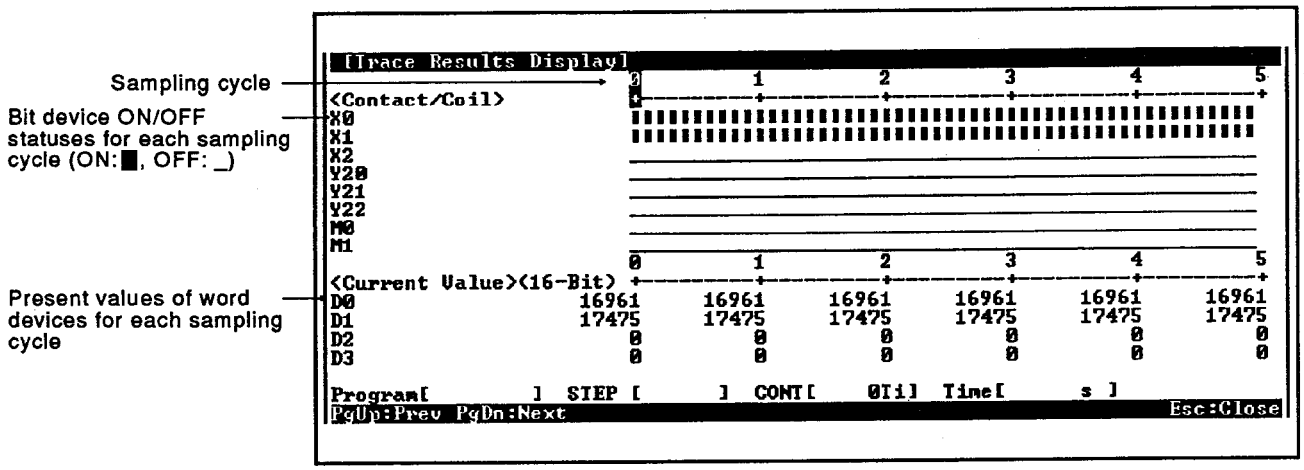


Fig. 8.1 Sampling Trace Operation

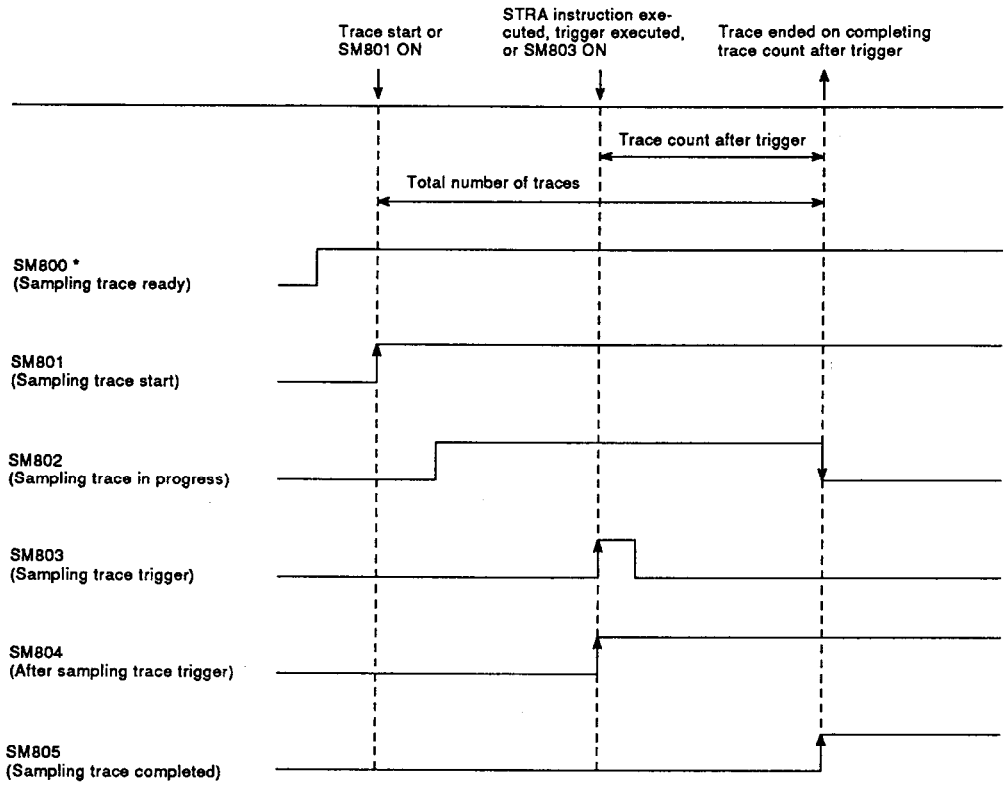
(d) The trace results show the ON/OFF statuses of bit devices, and present values of word devices, for each sampling cycle.



(2) Basic operation

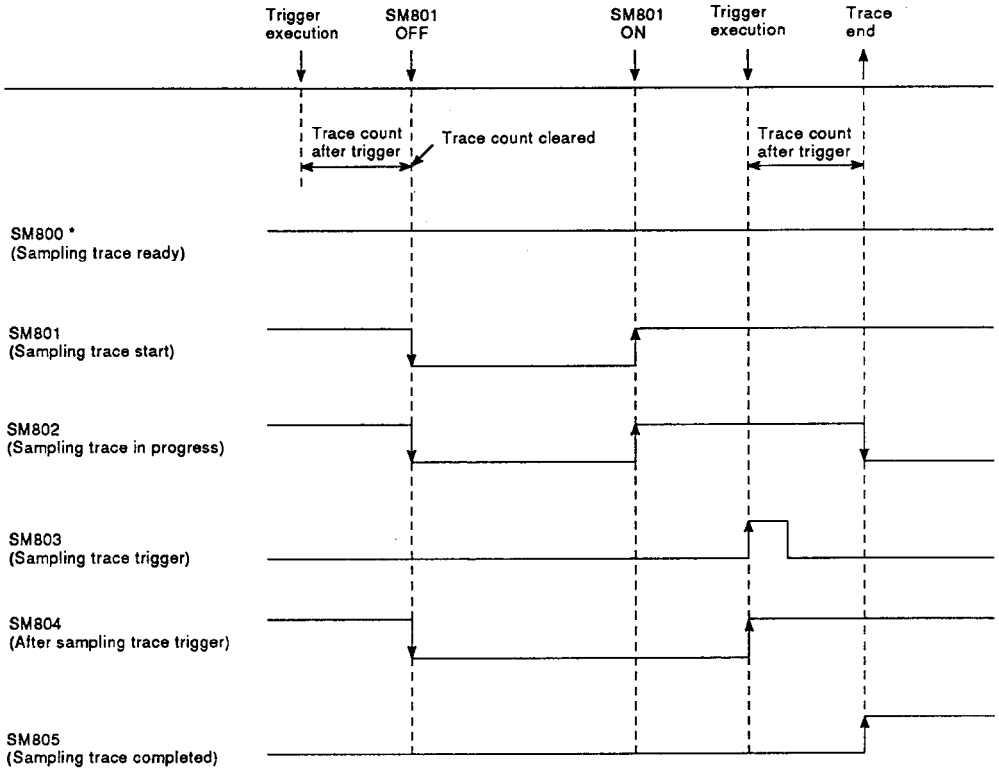
The basic operation for sampling trace is shown below.
The statuses during execution of the sampling trace function can be confirmed by monitoring special relays SM800 to SM805 and SM826.

- Trace execution



* SM800 comes ON automatically when preparation for sampling trace is completed.

- Suspending the trace



* When the trace is suspended from a peripheral device capable of GPP functions, SM800 is turned OFF.

The operation when an error occurs is as follows.
When an error occurs during sampling trace, SM826 (sampling trace error) comes ON, and at the same time, SM801 (sampling trace start) goes OFF.
To turn SM826 OFF, start the trace again.

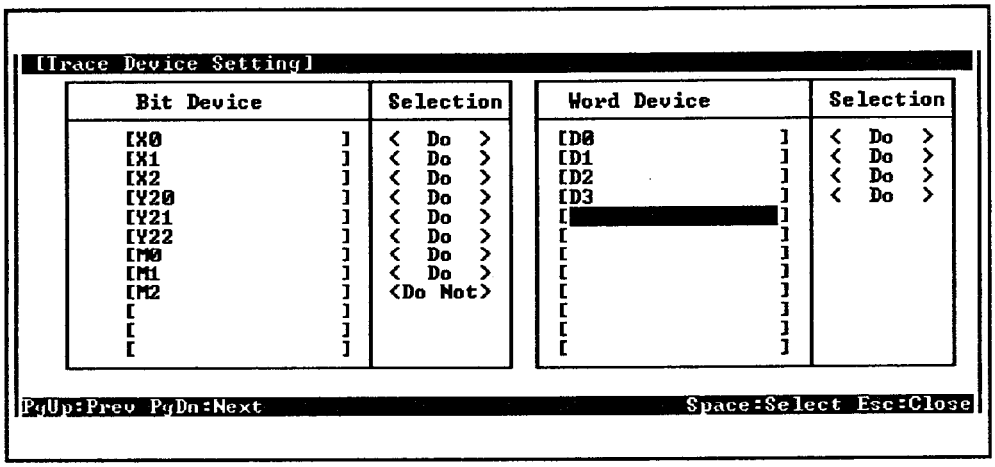
Operating Procedure

The operation used to perform sampling trace is described below.
These operations are performed on the "Sampling Trace" screen of the trace menu in the online mode.

(1) Set the trace devices and trace conditions with GPPQ.

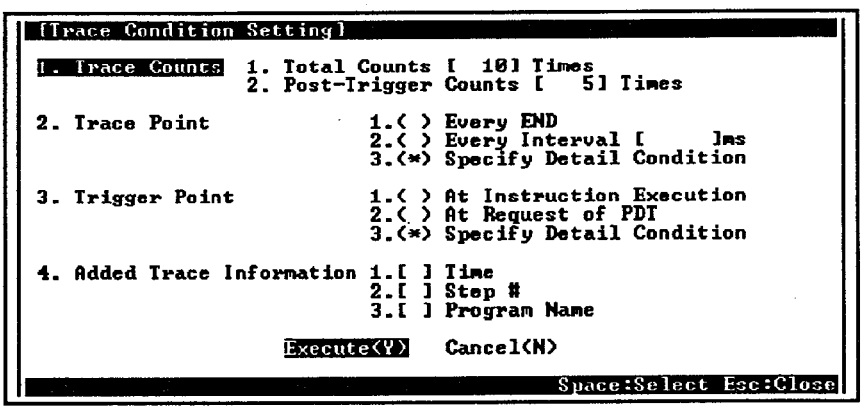
(a) Setting the trace devices

Set the devices at "Trace Device Setting" on the "Sampling Trace" screen.

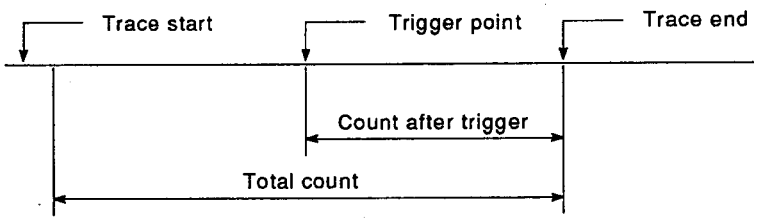


(b) Setting the trace conditions

Set the trace conditions at "Trace Device Setting" on the "Sampling Trace" screen.



Sampling stops after sampling the designated number of times (count after trigger) after execution of the trigger point.



The following is an explanation of the screen above:
One of the following four settings can be made for the trace condition: "1. Trace Counts", "2. Trace Point", "3. Trigger Point", or "4. Added Trace Information".

1) "Trace Counts"
For the total count, set the number of sampling traces executed between the trace start and trace end.
For the count after the trigger, set the number of sampling traces executed between execution of the trigger and the trace end.
The settings made for these counts must comply with the following formula:

Count after trigger ≤ total count ≤ 8192

2) "Trace Point"
Set the timing for collection of trace data. Select one of the following:

(a) Every END : Data collected at END instruction of every scan.

(b) Every Interval : Data collected at designated time. Set in 5 ms units in the range 5 to 5000 ms.

(c) Specify Detail Condition: Set a device and step number. Setting examples are presented below. The details on how to make the settings and data collection timing are the same as described in Section 8.2 for monitor condition setting for the monitor function.

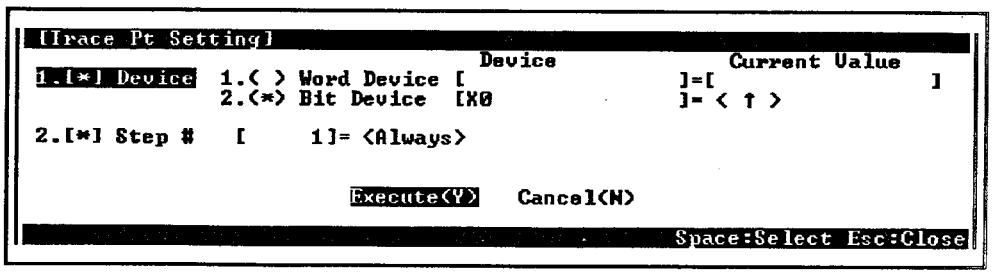
The following qualifications are possible with respect to the devices listed below.

(a) Bit devices : X, FX, Y, FY, M, L, F, SM, V, B, SB, T (contact), ST (contact), C (contact) J□\X, J□\Y, J□\B, J□\SB, BL□\S

(b) Word devices: T (present value), ST (present value), C (present value), D, SD, FD, W, SW, R, Z, ZR, U□\G, J□\W, J□\SW

The following qualifications are possible with respect to the devices listed above.

- Digit designation for bit devices
- Bit number designation for word devices

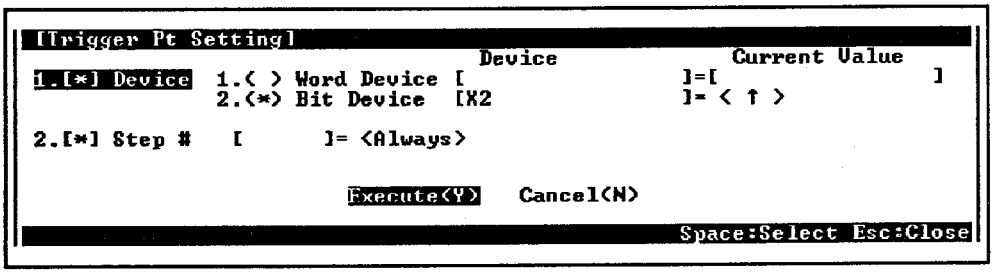


3) "Trigger Point"
Set the point at which the trigger is executed. Select one of the following:

(a) At Instruction Execution: Sets execution of the STRA instruction as the trigger.

(b) At Request of PDT : Sets trigger operation from a peripheral device capable of GPP functions as the trigger.

(c) Specify Detail Condition: Set a device and step number. Setting examples are presented below. The details on how to make the settings and trigger execution timing are the same as described in Section 8.2 for monitor condition setting for the monitor function.



4) "Added Trace Information"
Set information to be added at each trace. One or more of the following settings can be made (it is also possible to make no selection).

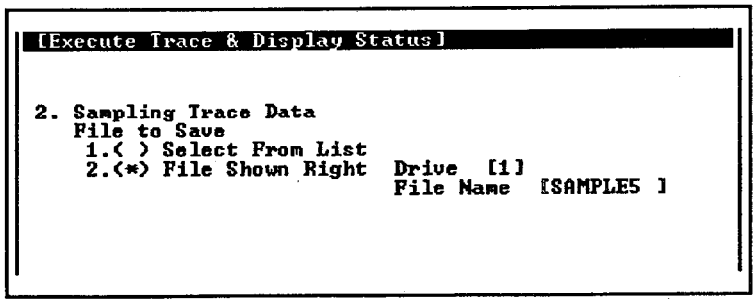
(a) Time : The time at which the trace was executed is stored.

(b) Step # : The step number at which the trace was executed is stored.

(c) Program Name: The program name for which the trace was executed is stored.

(2) Write the set trace device and trace condition to the memory card.

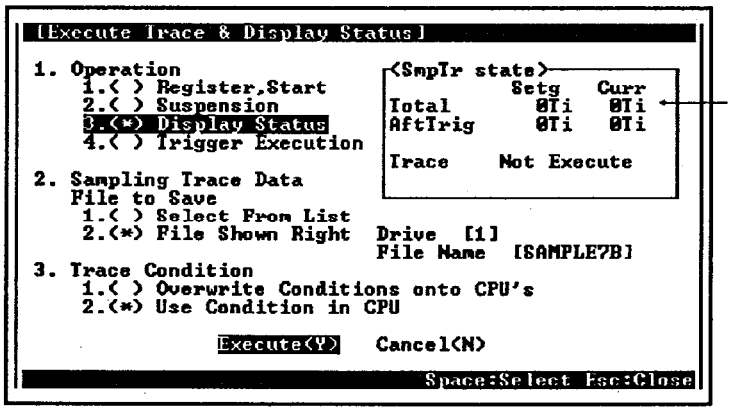
(a) Set the trace file and storage destination.
Set the drive number and file name at "1. ( ) Execute Trace & Display Status" on the "Sampling Trace" screen.



(b) Write the trace file to the memory card.
Write the trace file to the memory card by using "9. ( ) Write to PC (Condition)" on the "Sampling Trace" screen.
Since file names are used when writing to the memory card, multiple trace files can be written.

(3) Execute the sampling trace.

Execute the sampling trace by using "1. ( ) Execute Trace & Display Status" on the "Sampling Trace" screen.
A setting example for "1. ( ) Execute Trace & Display Status" is shown below.



The following is an explanation of the screen above:
The following settings can be made for "Execute Trace & Display Status" : "1. Operation", "2. Sampling Trace Data", and "3. Trace Condition".

(a) "Operation"

Select one of the following:

1) Register, Start : Registers and starts the trace. Starts the trace count.

2) Suspension : Suspends the trace. Clears the trace count and the count after the trigger. (To restart the trace, select "Register, Start" again.)

3) Display Status : Displays the trace statuses on the same screen as a message.

4) Trigger Execution: Starts the count after the trigger.
The trace is ended on reaching the designated count after the trigger.

(b) "Sampling Trace Data"

Select one of the following:

1) Select From List : Select from among the sampling trace files in the memory card.

2) File Shown Right: Set the drive number and sampling trace file name.

(c) "Trace Condition"

Select one of the following:

1) Overwrite Conditions onto CPU's: Overwrite the trace condition in an existing trace file.

2) Use Condition in CPU : Execute under the condition in the trace file designated in "2. Sampling Trace Data".

(4) Call the trace results from the CPU and display them.

(a) Read the trace results from the CPU by using "4. ( ) Read from PC (Results)" on the "Sampling Trace" screen.

(b) Display the trace results by using "4. ( ) Trace Results Display" on the "Sampling Trace" screen.

POINT

Once the sampling trace has been executed it is not executed a second time. To execute the trace again, execute the STRAR instruction to reset sampling trace.

Cautions

(1) Set sampling trace files in the RAM area of the memory card.

(2) It is possible to execute sampling trace from another station in the network, or from a serial communication module. However, sampling trace cannot be executed from more than one site at the same time. With the QnACPU, sampling trace can only be executed from one site at a time.

(3) Since the trace conditions registered in the CPU are latched, the trace data is retained even when the power is turned OFF. The data can be cleared by performing a latch clear operation using the RUN/STOP key switch on the QnACPU.

(4) The QnACPU must be connected to the peripheral device capable of GPP functions in order to execute sampling trace.

8.6 Status Latch Function

This function collects the data of devices at the designated point in time.

Application

Used to retain the statuses of devices used in a program at the designated point in time during debugging.

Function Description

(1) Function

(a) Status latch stores the device statuses at the designated point in time in a status latch file in a memory card.

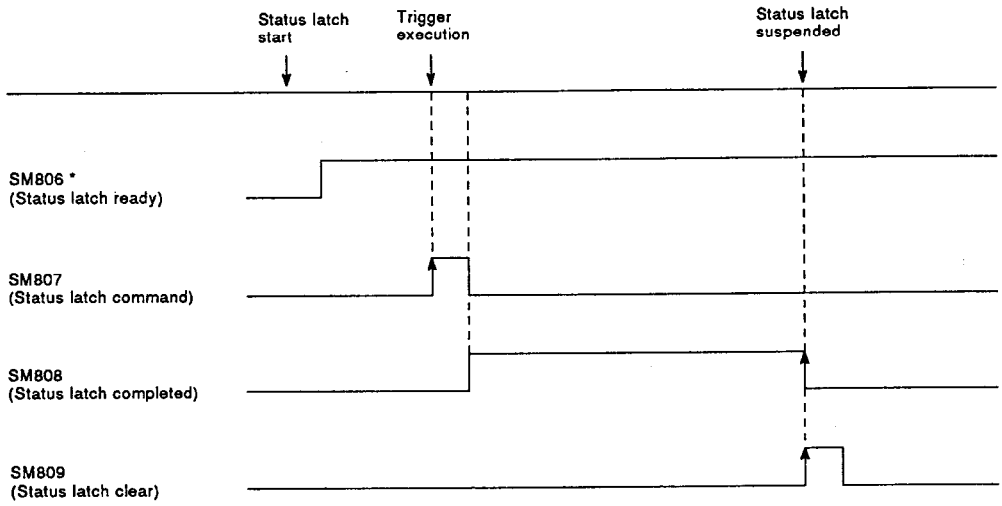
(b) The status latch file stores the status latch condition and status latch execution data for execution of status latch. Data is collected in response to one of the following conditions: execution of the SLT instruction in a program, execution of a status latch start at a peripheral device capable of GPP functions, or device and step number designation.

(c) The status latch results show the bit device ON/OFF statuses and word devices values at the designated point in time.

POINT

A memory card is required in order to execute status latch.

(2) Basic operation
The basic operation for status latch is shown below.
The statuses during execution of the status latch function can be checked by monitoring special relays SM806 to SM809 and SM827.



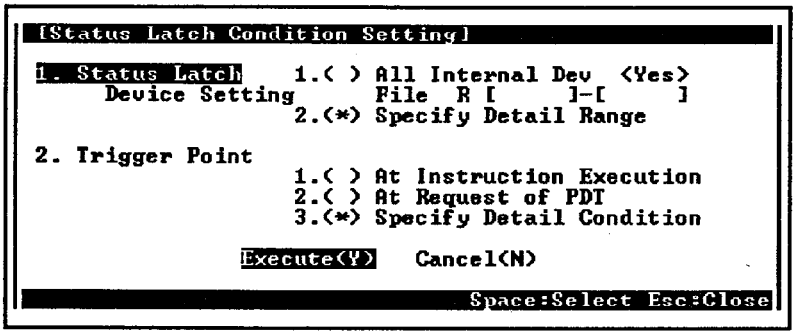
* SM806 comes ON automatically when preparation for status latch is completed.

(3) The operation when an error occurs is as follows.
When an error occurs during status latch, SM827 comes ON, and at the same time SM808 (completed) is turned ON.
To turn SM827 OFF, either turn SM809 ON or execute the SLTR instruction.

Operating Procedure

The operation used to perform status latch is described below.
These operations are performed on the "Status Latch" screen of the trace menu in the online mode.

(1) Setting the status latch condition
Set the status latch condition at "2. ( ) Status Latch Condition Setting" on the "Status Latch" screen.



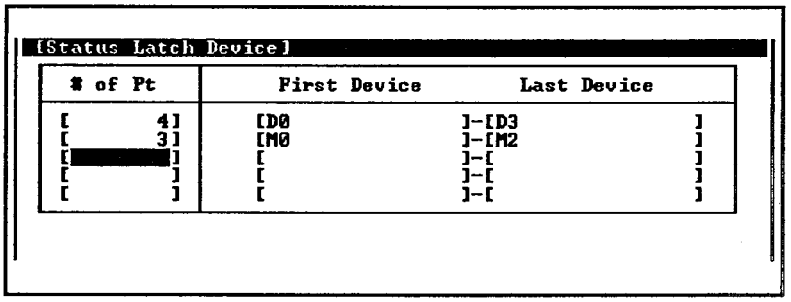
The following is an explanation of the screen above:
Either "1. Status Latch Device Setting" or "2. Trigger Point" can be set for the status latch condition setting.

(a) "Status Latch Device Setting"

Set the devices for which status latch is to be executed here. Select one of the following:

1) All Internal Dev : Set whether or not all internal devices are to be latched.

2) Specify Detail Range: Set the device types and numbers of points.
Setting examples are presented below.



(Applicable devices)

1) Bit devices : X, Y, M, L, F, SM, V, B, SB,
T (contact), T (coil), C (contact), C (coil),
ST (contact), ST (coil),
J□\X, J□\Y, J□\B, J□\SB, BL□\S

2) Word devices: T (present value), ST (present value),
C (present value), D, SD, W, SW, R, ZR,
U□\G, J□\W, J□\SW

REMARK

The maximum range of devices that can be set is 1000, including both bit devices and word devices. Qualification of the devices listed above is not possible.

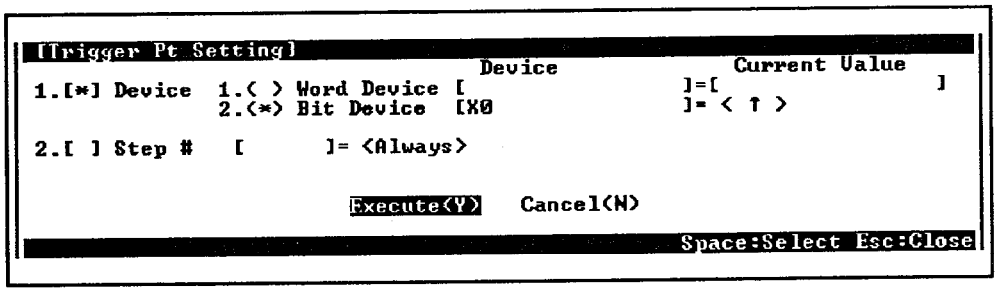
(b) "Trigger Point"

Set the condition under which the status latch is to be executed here. Select one of the following:

1) At Instruction Execution: Execution of the SLT instruction is made the trigger.

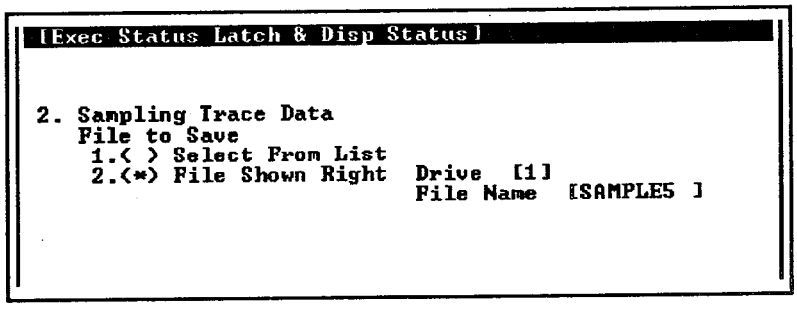
2) At Request of PDT : Sets trigger operation from a peripheral device capable of GPP functions as the trigger.

3) Specify Detail Condition: Set a device and step number. Setting examples are presented below. The details on how to make the settings and trigger execution timing are the same as described in Section 8.2 for monitor condition setting for the monitor function.



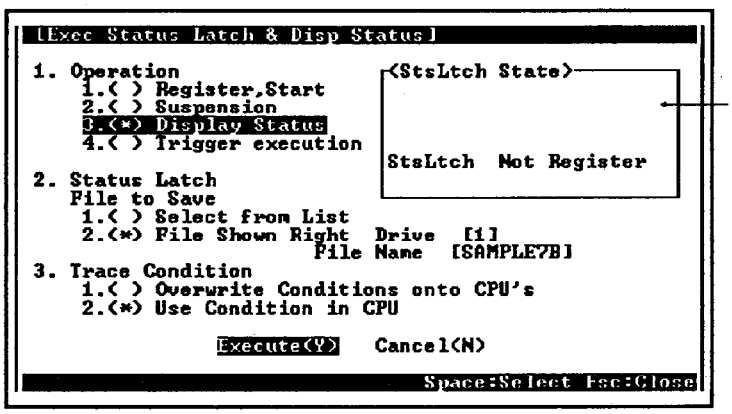
(2) Write the created status latch condition to the memory card.

(a) Set the status latch file and storage destination.
Set the status latch condition at "1. ( ) Exec Status Latch & Disp Status" on the "Status Latch" screen.



(b) Write the status latch file to the memory card.
Write the status latch file to the memory card using "7. ( ) Write to PC (Condition)" on the "Status Latch" screen.
Since file names are used when writing to the memory card, multiple status latch files can be written.

(3) Execute the status latch.
Execute the status latch by using "1. ( ) Exec Status Latch & Disp Status" on the "Status Latch" screen.
A setting example for "1. ( ) Exec Status Latch & Disp Status" is shown below.



The following is an explanation of the screen above:
The following settings can be made for "Exec Status Latch & Disp Status": "1 Operation", "2 Status Latch", and "3 Trace Condition".

(a) "Operation"

Select one of the following:

1) Register, Start : Registers and starts the status latch. Starts device data collection.

2) Suspension : Clears the status latch statuses.

3) Display Status : Displays the status latch statuses on the same screen as a message.

4) Trigger execution: Executes the trigger.

(b) "Status Latch"

Select one of the following:

1) Select from List : Select one of the status latch files in the memory card.

2) File Shown Right: Set the drive number and status latch file name.

(c) "Trace Condition"

Select one of the following:

1) Overwrite Conditions onto CPU's: Overwrite the status latch condition in an existing status latch file.

2) Use Condition in CPU : Execute under the condition in the status latch file designated in "2. Status Latch".

(4) Call the status latch results from the CPU and display them.

(a) Read the status latch results from the CPU by using "8. ( ) Read from PC (Results)" on the "Status Latch" screen.

(b) Display the read trace results by setting "1. ( ) Monitor Target" on the "Monitor Target Setting" screen of the "Option" menu in the ladder mode to "3. ( ) Status Latch".

Cautions

(1) Set status latch files in the RAM area of the memory card.

(2) It is possible to execute status latch from another station in the network, or from a serial communication module. However, sampling trace cannot be executed from more than one site at the same time. With the QnACPU, sampling trace can only be executed from one site at a time.

(3) Since the status latch conditions registered in the CPU are latched, the status latch data is retained even when the power is turned OFF. The data can be cleared by performing a latch clear operation using the RUN/STOP key switch on the QnACPU.

(4) The QnACPU must be connected to the peripheral device capable of GPP functions in order to execute sampling trace.

8.7 Step Run

This function runs a program one step at a time, runs only part of a program, or runs a program with a part skipped.

Application

Used to determine the causes of faults during debugging.

Function Description

This function can only be used when the CPU is set to STEP-RUN.
The step run function includes the following two functions. These functions are described in Sections 8.7.1 through 8.7.3.

- Step execution
- Partial execution

8.7.1 Step execution

Step execution is the execution of a sequence program one step at a time, starting from the designated step.
It allows execution of the sequence program while checking the execution of the sequence program and the contents of each device, for example during debugging.

There are two methods used for step execution:

(1) Step execution instruction by instruction

Instructions are executed one by one starting from the step where program operation is stopped. Program operation is stopped again after execution of each instruction.
This method is used to confirm the status of each devices after execution of one instruction.

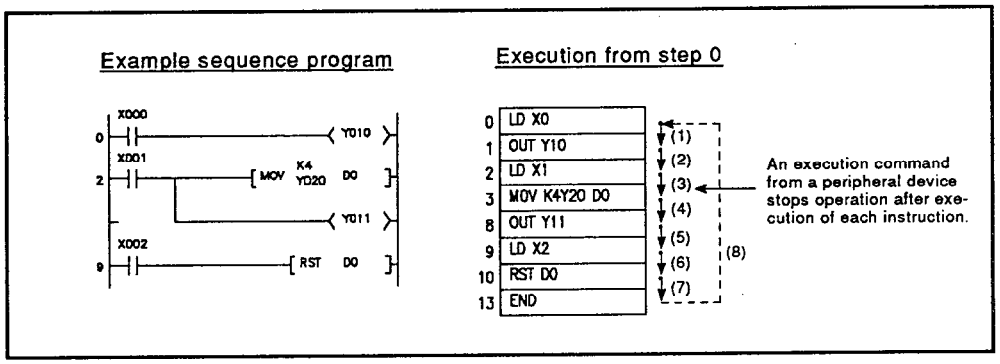


Fig. 8.2 Step Execution One Instruction at a Time

(2) Step execution with designated loop count

Program execution is repeated for the designated loop count (range: 1 to 32767) beginning with step 0 or the step where program operation was last stopped, and stops at the designated step.

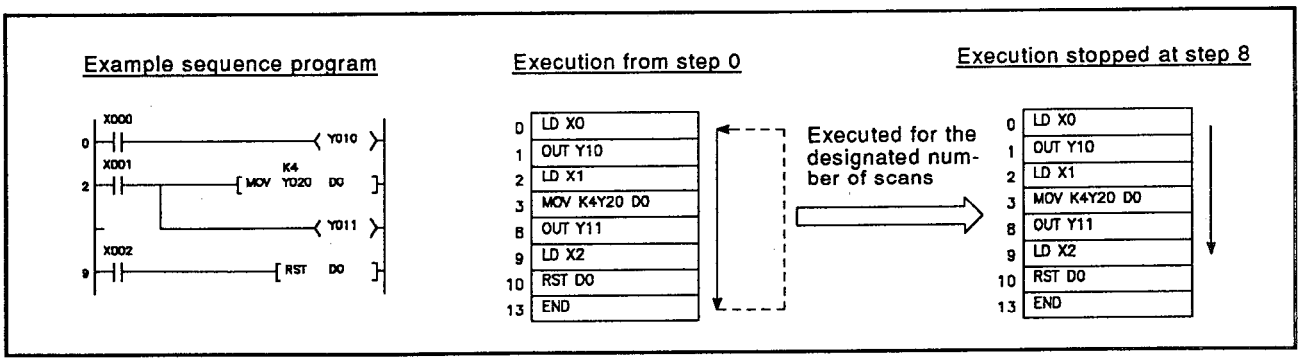
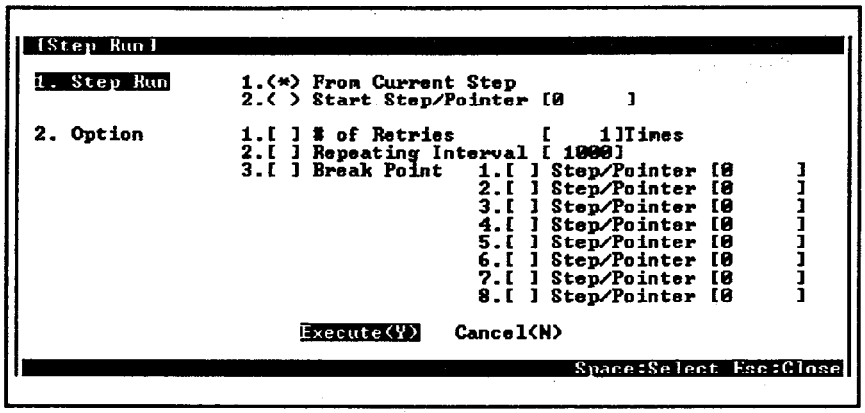


Fig. 8.3 Step Execution with Designated Loop Count

Operating Procedure

The operation used to perform step execution is described below.
These operations are performed on the monitor/test screen in the ladder mode (debugging).

(1) Select "B/Step Run"



8.7.2 Partial execution

The sequence program is executed from the start step or the step where operation is currently stopped and is stopped at a break point.

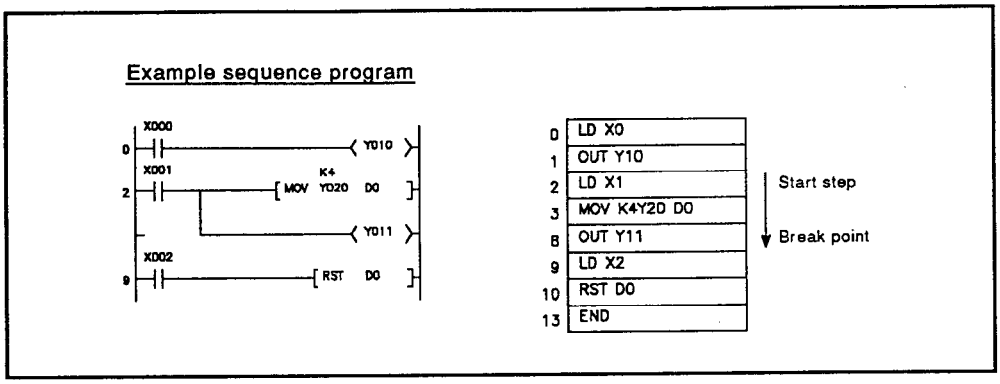


Fig. 8.4 Partial Execution

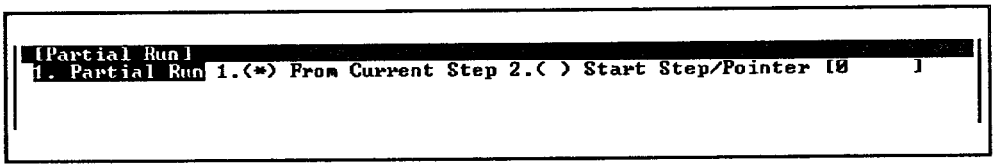
Operating Procedure

The operation used to perform partial execution is described below. These operations are performed on the monitor/test screen in the ladder mode (debugging).

(1) Designate the execution start step, break condition, and execution operation with GPPQ.

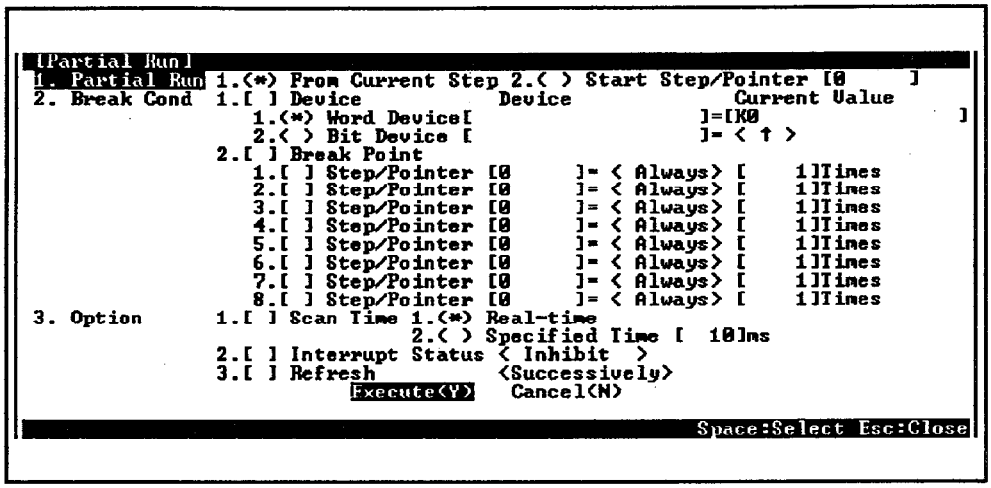
(a) Setting the execution start step

Designate the step at which partial execution is started at "1. Partial Run" on the "Partial Run" screen.



(b) Setting the break condition

Set the device status and break point at "2. Break Cond" on the "Partial Run" screen.



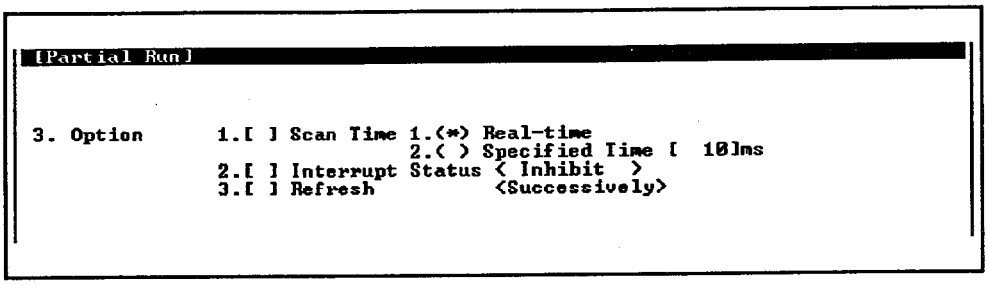
The devices that can be set are listed below.

1) Bit devices : X, FX, Y, FY, M, L, F, SM, V, B, SB

2) Word devices: T (contact, present value), ST (contact, present value), C (contact, present value), D, SD, FD, W, SW, R, ZR, U□\G, J□\X, J□\Y, J□\B, J□\SB, J□\W, J□\SW, BL□\S

(c) Setting the execution operation

Set the scan time, interrupt status, and refresh, at "3. Option" on the "Partial Run" screen.



* More than one setting can be made.

The settings are explained below.

Item	Operation (Setting) Details
Scan Time	Designates whether the scan time for the partial operation is to be the actual time taken or a designated time. (The default is a designated time of 10 ms.)
Interrupt Status	Designates whether or not interrupts are prohibited during execution. (Default: "Inhibit")
Refresh	Designates whether I/O refresh is executed every time program execution is stopped due to satisfaction of a condition, or only at END processing. (Default: "Successively")

8.7.3 Skip function

Skip execution or partial execution of a program whereby the program is executed with the designated step(s) skipped.

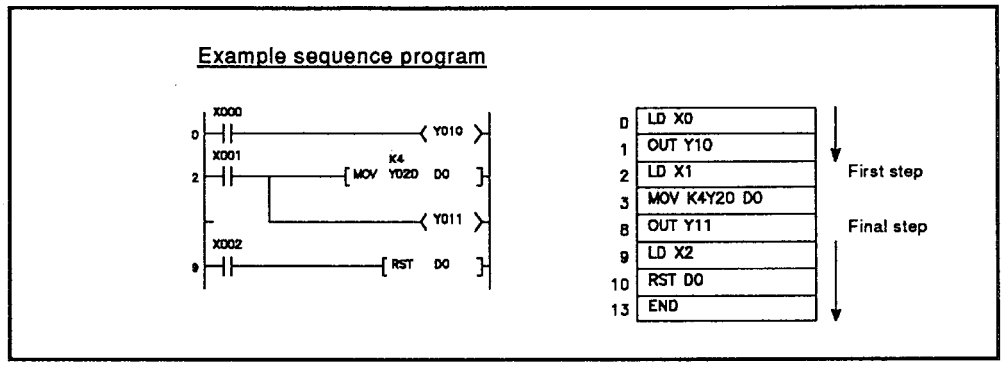
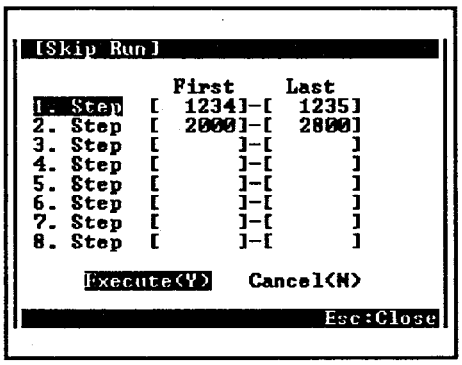


Fig. 8.5 Skip Function

Operating Procedure

The operation used to perform skip execution is described below.
These operations are performed on the monitor/test screen in the ladder mode (debugging).

(1) Set the program range to be skipped using GPPQ.
Designate the step number(s) to be skipped on the "D/Skip Run" screen.



8.8 Program Trace Function

This function collects program execution statuses.

Application

Used to check the execution status of any step of any program during debugging.
This shortens debugging time.

Function Description

(1) Function

(a) The program trace function collects the execution status of the designated step of the designated program and stores it in a program trace file in the memory card.

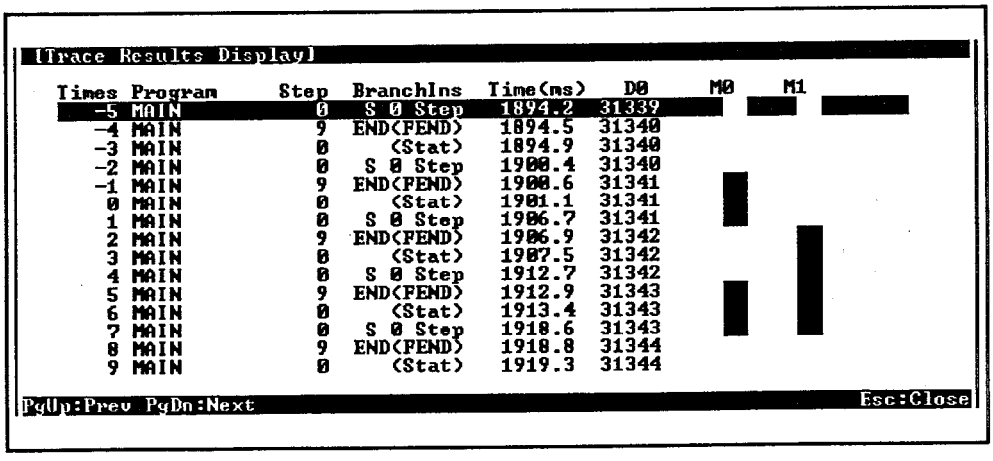
(b) The devices that can be traced are listed below.

1) Bit devices : X, FX, DX, Y, FY, DY, M, L, F, SM, V, B, SB, T (contact), T (coil), ST (contact), ST (coil), C (contact), C (coil), J□\X, J□\Y, J□\B, J□\SB, BL□\S . . . . . .Max. 50 points

2) Word devices: T (present value), ST (present value), C (present value), D, SD, FD, W, SW, R, Z, ZR, U□\G, J□\W, J□\SW . . . . . Max. 50 points

(c) The program trace file stores the trace condition data and trace execution data required to execute a program trace. After a trace has been started at a peripheral device capable of GPP functions, it is continued until the set number of traces is completed.

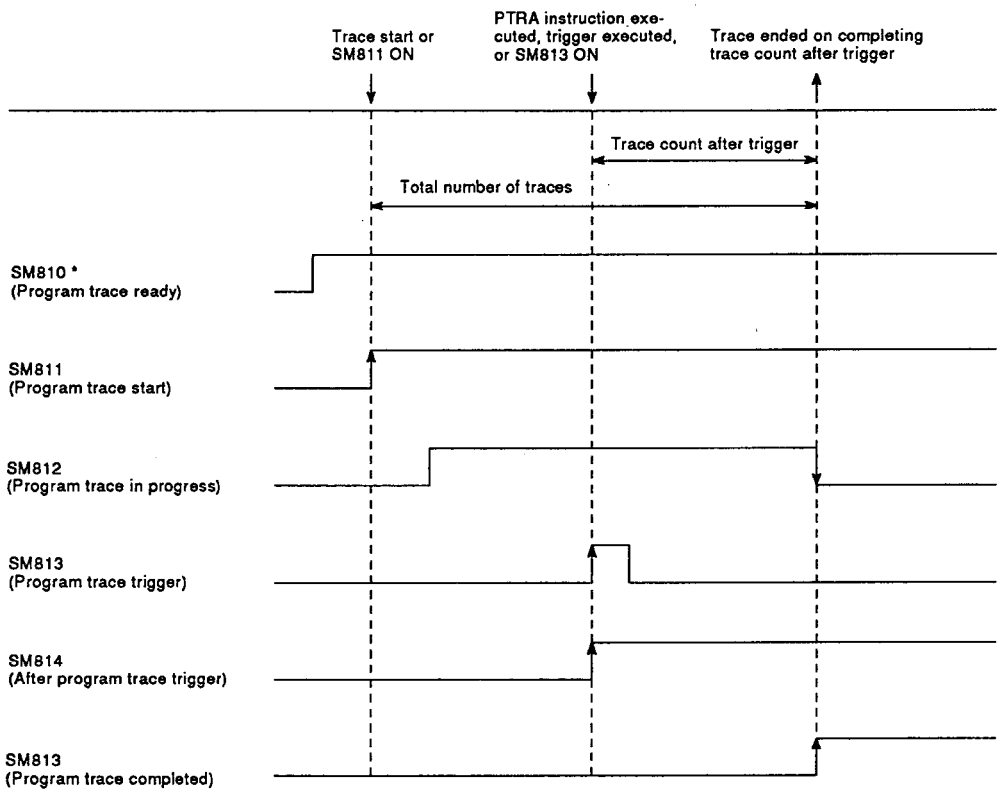
(d) The trace results show the program name, step No. device status, etc., for each trace No..



(2) Basic operation

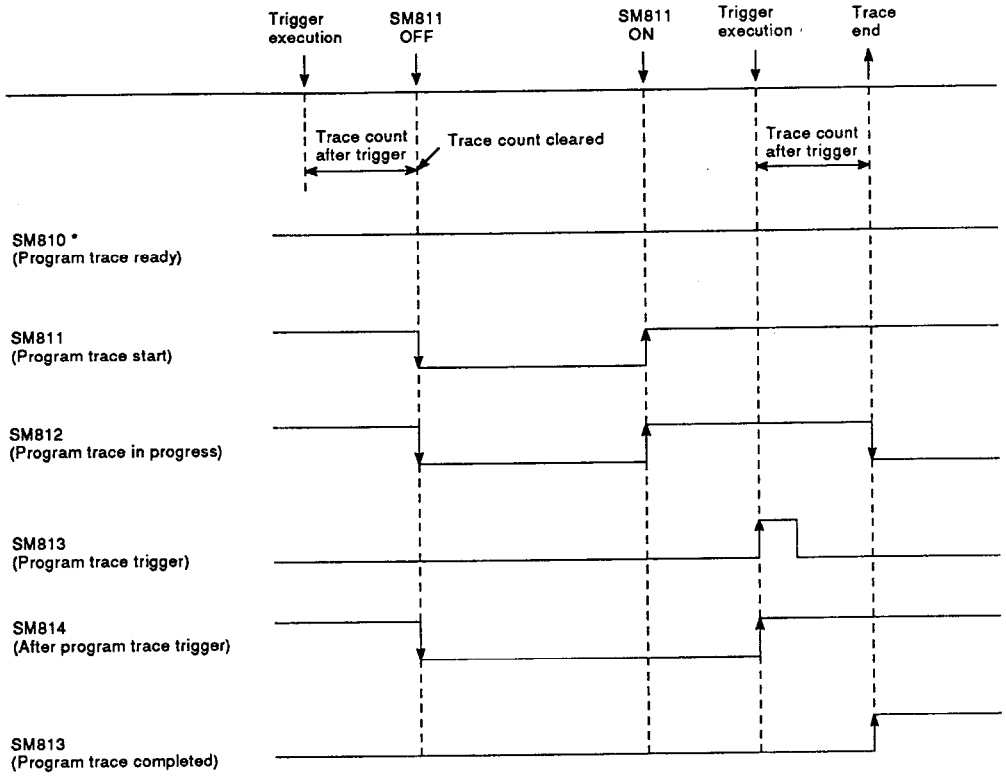
The basic operation for program trace is shown below.
The statuses during execution of the program trace function can be confirmed by monitoring special relays SM810 to SM815 and SM828.

- Without suspension of the trace



* When ready for program trace, SM810 is automatically turned ON.

- With trace suspension



* When the trace is suspended from a peripheral device capable of GPP functions, SM810 is turned OFF.

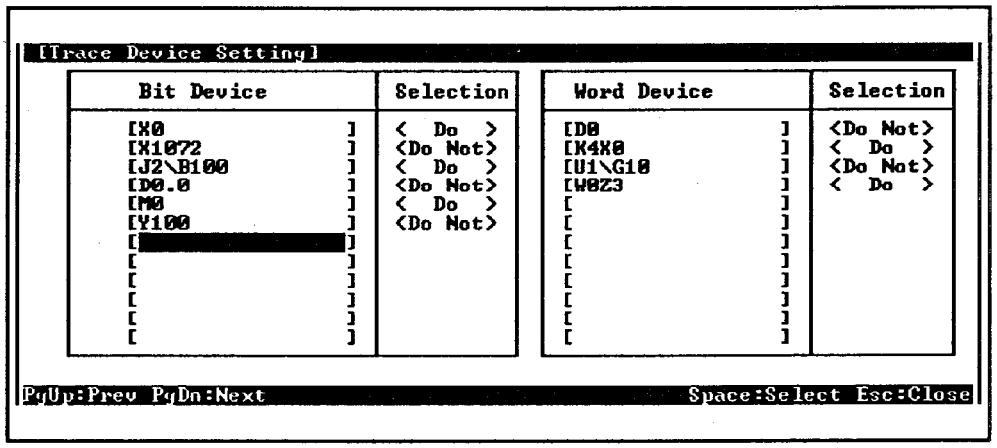
The operation when an error occurs is as follows.
When an error occurs during program trace, SM828 (program trace error) comes ON, and at the same time, SM811 (program trace start) goes OFF.
To turn SM828 OFF, either turn SM811 ON, or execute the PTRA instruction.

Operating Procedure

The operation used to perform program trace is described below.
These operations are performed on the "Program Trace" screen of the trace menu in the online mode.
Perform these operations with the CPU set to the STEP-RUN state (see Section 8.7).

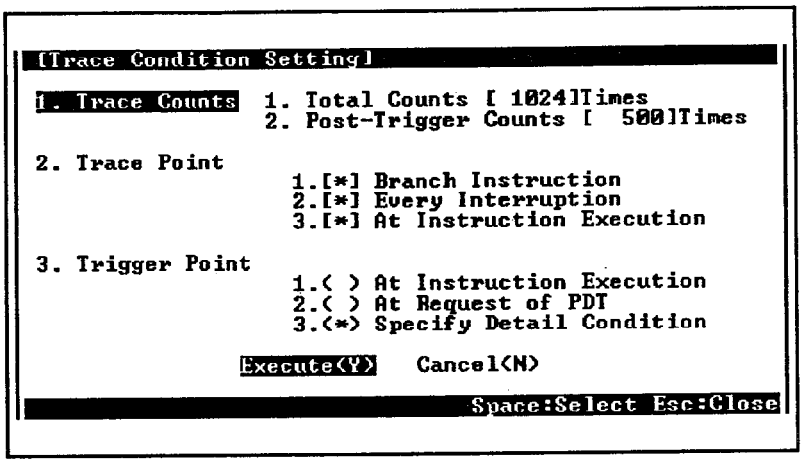
(1) Set the trace devices and trace conditions with GPPQ.

(a) Setting the trace devices
Set the devices at "Trace Device Setting" on the "Program Trace" screen.



(b) Setting the trace conditions

Set the trace conditions at "Trace Condition Setting" on the "Program Trace" screen.



The following is an explanation of the screen above:
One of the following three settings can be made for the trace condition: "1. Trace Counts", "2. Trace Point", or "3. Trigger Point".

1) "Trace Counts"
For the total count, set the number of sampling traces executed between the trace start and trace end.
For the count after the trigger, set the number of sampling traces executed between execution of the trigger and the trace end.
The settings made for these counts must comply with the following formula:

Count after trigger ≤ total count ≤ 8192

2) "Trace Point"
Set the point at which the trace is to be executed. One or more of the following settings can be made

(a) Branch Instruction : Executed at each CALL, JMP, or other instruction.

(b) Every Interruption : Executed at each interrupt program.

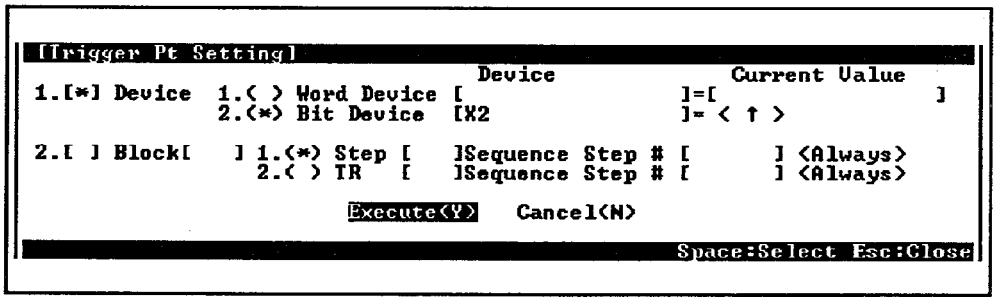
(c) At Instruction Execution: Executed at each PTRAEXE instruction.

3) "Trigger Point"
Set the point at which the trigger is executed. Select one of the following:

(a) At Instruction Execution: Sets execution of the PTRA instruction as the trigger.

(b) At Request of PDI : Sets trigger operation from a peripheral device capable of GPP functions as the trigger.

(c) Specify Detail Condition: Set a device and step number. Setting examples are presented below. The details on how to make the settings and trigger execution timing are the same as described in Section 8.2 for monitor condition setting for the monitor function.



The following qualifications are possible with respect to the devices listed below.

- Bit devices : X, FX, Y, FY, M, L, F, SM, V, B, SB, T (contact), ST (contact), C (contact), J□\X, J□\Y, J□\B, J□\SB, BL□\S
- Word devices: T (present value), ST (present value), C (present value), D, SD, FD, W, SW, R, Z, ZR, U□\G, J□\W, J□\SW

The following qualifications are possible with respect to the devices listed above.

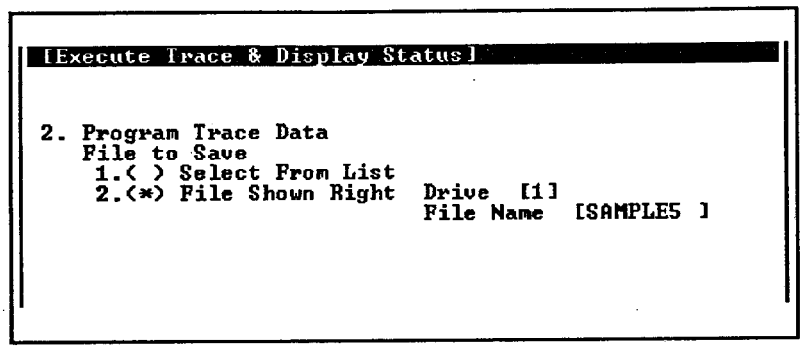
- Digit designation for bit devices
- Bit number designation for word devices

POINT

The time at which the trace was executed, program name, and step are automatically added to the trace data.

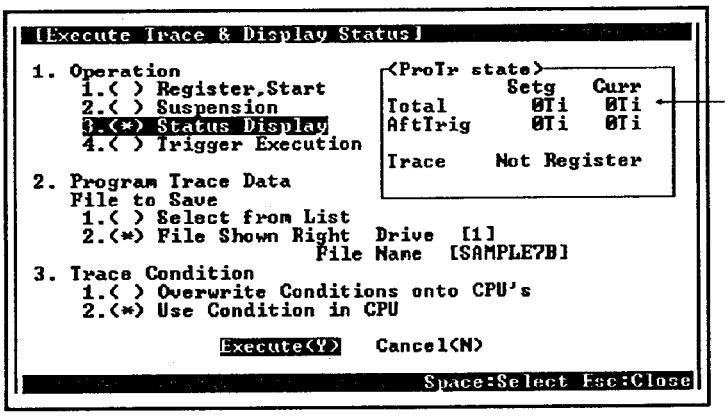
(2) Write the set trace device and trace condition to the memory card.

(a) Set the trace file and storage destination.
Set the drive number and file name at "1. ( ) Execute Trace & Display Status" on the "Program Trace" screen.



(b) Write the trace file to the memory card.
Write the trace file to the memory card by using "9. ( ) Write to PC (Condition)" on the "Program Trace" screen.
Since file names are used when writing to the memory card, multiple trace files can be written.

(3) Execute the program trace.
Execute the program trace by using "1. ( ) Execute Trace & Display Status" on the "Program Trace" screen.
A setting example for "1. ( ) Execute Trace & Display Status" is shown below.



Displayed only at status display selection.

The following is an explanation of the screen above:
The following settings can be made for "Execute Trace & Display Status": "1. Operation", "2. Program Trace Data", and "3. Trace Condition".

(a) "Operation"

Select one of the following:

1) Register, Start : Starts the trace. Starts the trace count.

2) Suspension : Suspends the trace. Clears the trace count and the count after the trigger. (To restart the trace, select "Register, Start" again.)

3) Status Display : Displays the trace statuses on the same screen as a message.

4) Trigger Execution: Starts the count after the trigger. The trace is ended on reaching the designated count after the trigger.

(b) "Program Trace Data"

Select one of the following:

1) Select from List : Select from among the program trace files in the memory card.

2) File Shown Right: Set the drive number and program trace file name.

(c) "Trace Condition"

Select one of the following:

1) Overwrite Conditions onto CPU's: Overwrite the trace condition in an existing trace file.

2) Use Condition in CPU : Execute under the condition in the trace file designated in "2. Program Trace Data".

(4) Call the trace results from the CPU and display them.

1) Read the trace results from the CPU by using "A. ( ) Read from PC (Results)" on the "Program Trace" screen.

2) Display the read trace results by using "4. ( ) Trace Results Display" on the "Program Trace" screen.

POINT

Once the program trace has been executed it is not executed a second time. To execute the trace again, execute the PTRAR instruction to reset program trace.

Cautions

(1) Set program trace files in the RAM area of the memory card.

(2) It is possible to execute program trace from another station in the network, or from a serial communication module. However, sampling trace cannot be executed from more than one site at the same time. With the QnACPU, sampling trace can only be executed from one site at a time.

(3) The QnACPU must be connected to the peripheral device capable of GPP functions in order to execute sampling trace.

(4) Program trace can only be executed when the CPU is set to the STEP-RUN state.

8.9 Simulation Function

Application

This function simulates execution of a program in step execution or partial execution, with the input module, output module, or special function module isolated from the CPU. This makes it possible to perform program debugging without any effects on other modules.

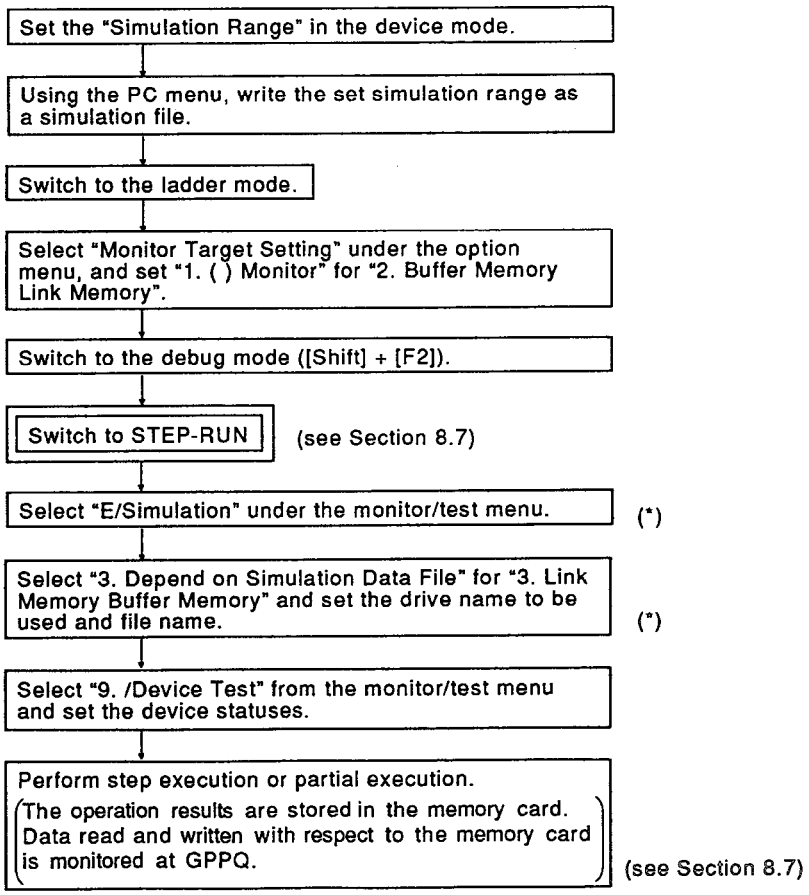
Function Description

(1) When the program is executed, isolation from inputs from external sources and outputs to external destinations is achieved by making a setting so that input refresh and output refresh processing is not performed for the input modules and output modules.

(2) Isolation from special function module operations is achieved by setting "Ignore" or "Depend on Simulation Data File" with respect to the buffer memory of the special function module.

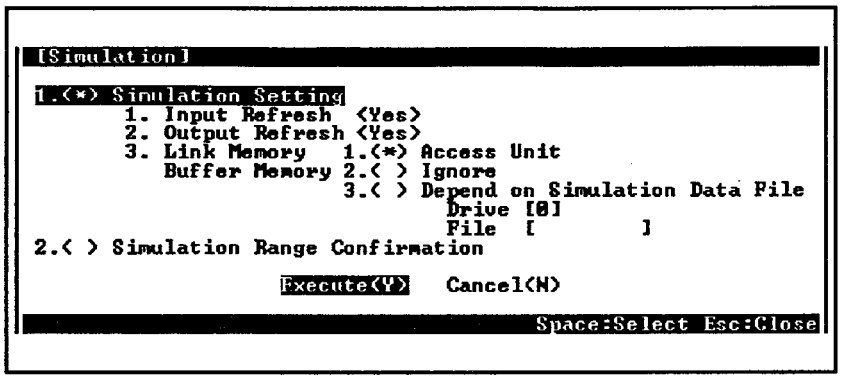
Operating Procedure

The operation used to perform simulation is described below.
[ ] indicates a GPPQ operation and [ ] indicates and operation at the CPU.



(*) Described in this section.

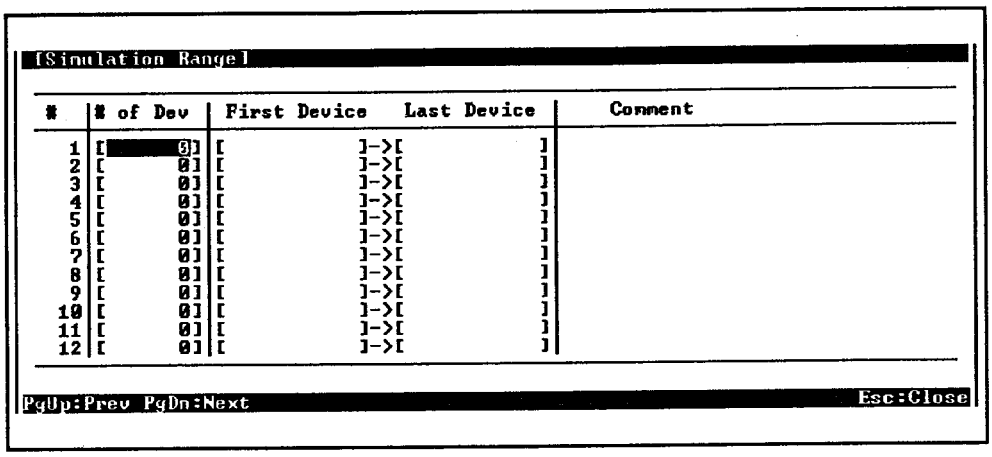
(1) Make the settings on the simulation setting screen shown below.



- Details on the settings that can be made for each item are given below:

Item	Setting Options	Description
Input Refresh	Yes/No	Select whether inputs from external sources are input to the CPU or not.
Output Refresh	Yes/No	Select whether the operation results in the CPU are output to external destinations or not.
Link Memory Buffer Memory	Access Unit Ignore Depend on Simulation Data File	Select the method of accessing each module.

If "Depend on Simulation Data File" is selected for "Link Memory Buffer Memory", the access range for each module can be checked by checking the simulation range settings.



Cautions

(1) The CPU must be set to STEP-RUN.

(2) A memory card is required to carry out link memory/buffer memory simulation using a simulation data file.

(3) It is possible to carry out simulation from another station in the network, or from a serial communication module. However, simulation cannot be executed from more than one site at the same time.
With the QnACPU, sampling trace can only be executed from one site at a time.

(4) The QnACPU must be connected to the peripheral device capable of GPP functions in order to execute sampling trace.

(5) Note the following points when executing simulation:

- If direct inputs (DX) and direct outputs (DY) are used to handle inputs/outputs directly, the device memory is accessed rather than the actual inputs/outputs.
- No processing is performed for any special function module instruction.
- When a SP UNIT ERROR occurs, FFFFH is displayed in the module number area of the common information.
- If "Ignore" is set for the buffer memory access method, FFFFH is set for access by instruction and the monitor results.

8.10 Debugging by Several People

This function allows simultaneous debugging from several peripheral devices capable of GPP functions.

Application

Used to simultaneously debug different files from more than one peripheral device capable of GPP functions.

Function Description

The combinations of debugging functions that can be used simultaneously by different operators are indicated in the table below.

	Monitoring	Write during RUN	Execution Time Measurement	Sampling Trace /Program Trace	Status Latch	Step Run	Simulation
Monitoring	o	x	o	o	o	o	o
Write during RUN	x	o	x	x	x	x	x
Execution time measurement	o	x	x	o	o	o	o
Sampling trace	o	x	o	x	o	o	o
Program trace	o	x	o	x	o	o	o
Status latch	o	x	o	o	x	o	o
Step run	o	x	o	o	o	x	o
Simulation	o	x	o	o	o	o	x

o: Simultaneous execution possible (but the detailed condition setting at only one peripheral device capable of GPP functions is valid; detailed conditions cannot be set at the other peripheral devices capable of GPP functions).

x: Can only be executed from one peripheral device capable of GPP functions.

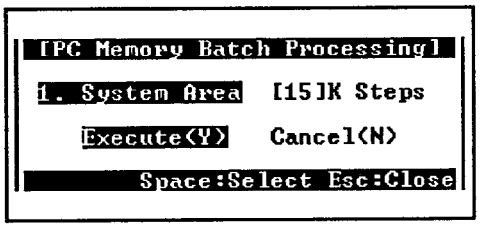
8.10.1 Simultaneous monitoring by more than one person

This function allows several people to carry out monitoring simultaneously. This enables high-speed monitoring by setting monitoring files for other stations (monitor file setting at the host station is not required).

Operating Procedure

The operation for simultaneous monitoring by more than one person is described below.

(1) Select "5. ( ) Format (with Option)" for "B/PC Memory Batch Processing" in the "2/PC" menu in the online mode, and set a monitor file for another station.
Setting examples are presented below.



Up to 15 k steps in 1 k step units can be set as the system area.
The area corresponding to one monitor file for another station is no more than 1 k steps.
Accordingly, a maximum of 15 monitor files can be set.
Since the program area occupies the same area as the monitor files for other stations, the program area is reduced by the area set for monitor files for other stations.

Cautions

(1) The detailed conditions for monitoring can be set from one site only.

(2) Monitoring from other stations is possible without setting monitor files for other stations, but in this case high-speed monitoring will not be possible.

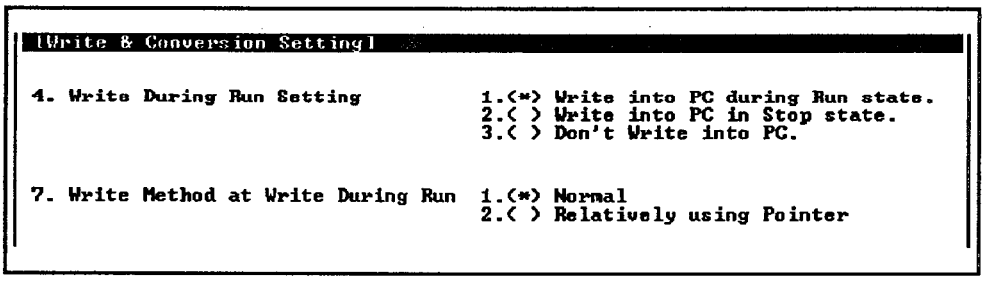
8.10.2 Simultaneous execution of write during RUN by more than one person

This function allows several people to perform write during RUN operations simultaneously with respect to the same file or different files.

Operating Procedure

The operation for simultaneous write during RUN executed by more than one person is described below.

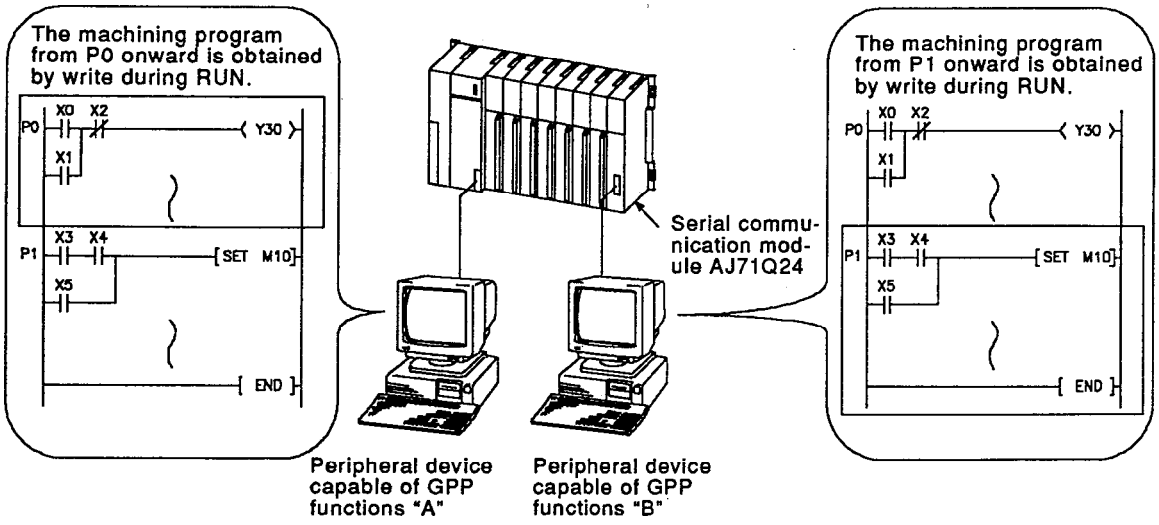
(1) Set "4. Write During Run Setting" or "7. Write Method at Write During Run" for "4/Write & Conversion Setting" in the "Option" menu in the ladder mode.
Setting examples are presented below.



(a) Set "1. ( ) Write into PC during Run state." for "4. Write During Run Setting".

(b) Select "1. ( ) Normal" or "2. ( ) Relatively using Pointer" for "7. Write Method at Write During Run".

If more than one person is to perform a write during RUN operation with respect to the same file, set a write during RUN pointer in advance and select "2. ( ) Relatively using Pointer".
The example below shows a case where peripheral device capable of GPP functions "A" performs write during RUN from P0, and peripheral device capable of GPP functions "B" performs write during RUN from P1. The program enclosed in the frame ☐ is the program subject to write during RUN.



Cautions

See Section 8.2.

9. MAINTENANCE FUNCTIONS

9.1 Function List

The functions for maintenance are listed below.

Item		Description	Refer to
Watchdog timer		Function that monitors watchdog errors due to CPU hardware or program errors.	Section 9.2
Self diagnosis		Function whereby the QnACPU itself diagnoses whether or not there are any errors.	Section 9.3
Fault history		Function that stores the results of diagnosis in memory as a fault history.	Section 9.4
System protect		Function that sets whether reading/writing is enabled or disabled for QnACPU files.	Section 9.5
Password registration		Function that disables GPPQ operations with respect to the CPU.	Section 9.6
Online I/O module change		Function that allows replacement of I/O modules while the CPU is in the RUN state.	Section 9.7
System display		Function that allows monitoring of the system configuration by connecting a peripheral device capable of GPP functions.	Section 9.8
LED/LED indicator indications		Function that indicates the operating state of the CPU by means of LEDs or the LED indicator on the front face of the CPU module.	Section 9.9
	LED	Indicates whether CPU operation is normal or abnormal.	Section 9.9.1
	LED indicator	Displays a message when an error occurs.	Section 9.9.2

For details on operations at GPPQ, refer to the SW□IVD-GPPQ Operating Manual (Online).

9.2 Watchdog Timer

(1) Watchdog timer

The watchdog timer is an internal timer in the programmable controller to detect programmable controller hardware errors and sequence program errors.
The default setting for this timer is 200 ms.

REMARK

The time set for the watchdog timer can be changed using "WDT" in PC RAS setting in the GPPQ parameter mode.
The setting range is 10 to 2000 ms (in 10 ms units).

(2) Resetting the watchdog timer

The QnACPU resets the watchdog timer during END processing.
When the QnACPU is operating normally and the END instruction is executed in the sequence program within the set value of the watchdog timer, the watchdog timer does not time out. It times out when the END instruction is not executed within the value set for the watchdog timer due to a QnACPU hardware error or an excessively long sequence program scan time.

REMARK

The "scan time" is the time taken for the execution of the sequence program, starting from step 0 and ending at step 0.
The scan time is not the same in every scan: it differs in accordance with the execution or non-execution of the instructions used in the program (see Section 12.1).

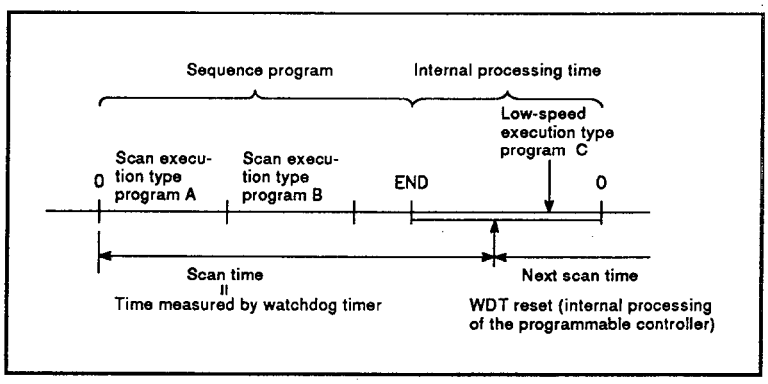


Fig. 9.1 Resetting the Watchdog Timer

(3) Processing when the watchdog timer times out

When the scan time exceeds the set value of the watchdog timer, a watchdog timer error occurs and the programmable controller operates as follows.

(a) All programmable controller outputs are turned OFF.

(b) The RUN LED on the front of the CPU module goes off and the ERROR LED flashes.

(c) SM1 turns ON and the error code is stored in SD0.

REMARK

The watchdog timer can be reset by a WDT instruction in the sequence program.
However, the scan time value is not reset and scan time is measured up to the END instruction.

POINT

An error will occur if the time set for the watchdog timer is less than 10 ms.

9.3 Self Diagnosis

The self-diagnosis function is a function whereby the QnACPU diagnoses its own errors.

(1) The self-diagnosis function serves to prevent malfunctions of the programmable controller, and to facilitate preventive maintenance.
Self-diagnosis processing is executed if an error occurs at QnACPU power ON or while the programmable controller is in the RUN state, and it involves the display of the error detected by the self-diagnosis function, stopping of programmable controller operation, etc.

(2) The QnACPU stores the error code corresponding to the error that has occurred in special register SD0, turns the ERROR LED on, and displays an error message on the LED indicator.
If more than one error has occurred, the error code of the latest error is stored in SD0.

(3) Even if the power is turned OFF, battery back-up retains a record of the latest 16 errors.
(See Section 9.4)
This record can be checked by means of the fault history function in the PC diagnosis mode.

(4) When an error is detected by self-diagnosis, CPU operation complies with one of the following two modes:

- Mode in which programmable controller operation is stopped
When an error is detected, operation is stopped immediately and all outputs (Y) are turned OFF.
- Mode in which programmable controller operation is continued
When an error is detected, only the program part affected by the error is not executed; the rest of the program is executed.

In addition, settings can be made in PC RAS setting in the parameter mode to continue operation or stop operation when the following errors occur:

1 Calculation (including SFC programs)

2 Extended Ins

3 Fuse Blown

4 I/O Unit Compare

5 Sp Unit Access

6 IC Card Access

7 IC Card Operate

(The default for all of the these in the parameters is "Pause".)

For example, if "Resume" is set for I/O module verify error, when this error occurs operation is continued using I/O address before the error occurred.

When an error is detected, a record of the error occurrence is stored in the special relays (SM0, SM1) and the error contents are stored in a special register (SD0). Use these special relays and this special register in the sequence program to establish programmable controller or mechanical system interlocks.

(5) It is possible to select whether or not the following checks are performed by setting "Yes/No" for error check in PC RAS setting in the parameter mode.

1 Battery Check

2 Fuse Blown Check

3 I/O Unit Compare

(The default for all of these in the parameter settings is "Yes".)

If "No" is set for error check, error detection is not performed for these items, which shortens the processing time for the END instruction.

Self-Diagnosis List

Diagnosis Details		Diagnosis Timing	CPU Status	LED Statuses		LED Indicator Message (Q3A/Q4ACPU Only)
				RUN	ERROR	
Hardware faults	CPU error	Constant check	Stop	OFF	Flashes	MAIN CPU DOWN
	END instruction not executed	When an END instruction is executed	Stop	OFF	Flashes	END NOT EXECUTE
	RAM check	• ON switching on the power or resetting	Stop	OFF	Flashes	RAM ERROR
	Operation circuit check	ON switching on the power or resetting	Stop	OFF	Flashes	OPE. CIRCUIT ERR.
	Blown fuse (Default ... stop) *1	When an END instruction is executed (Default ... check executed) *2	Stop	OFF/ON	Flashing/ON	FUSE BREAK OFF
	I/O allocation error	When an interruption occurs	Stop	OFF	Flashes	I/O INT. ERROR
	Special function module error	• On switching on the power or resetting • When a FROM/TO instruction is executed	Stop	OFF	Flashes	SP. UNIT DOWN
	Control bus error	• On switching on the power or resetting • When a FROM/TO instruction is executed	Stop	OFF	Flashes	CONTROL-BUS ERR.
	Occurrence of momentary power interruption	Constant check	Operation continues	ON	OFF	AC DOWN
	Battery voltage drop	Constant check (Default ... check executed) *3	Operation continues	ON	OFF	BATTERY ERROR

Self-Diagnosis List (Continued)

	Diagnosis Details	Diagnosis Timing	CPU Status	LED Statuses		LED Indicator Message (Q3A/Q4ACPU Only)
				RUN	ERROR	
Handling error	I/O module verification (Default ... stop) *1	When an END instruction is executed (Default ... check executed) *2	Stop /Operation continues	OFF/ON	Flashing/ON	UNIT VERIFY ERR.
	Special function module allocation error	• On switching ON the power or resetting • On switching from STOP to RUN	Stop	OFF	Flashes	SP. UNIT LAY. ERR.
	Special function module error (Default ... stop) *1	When a FROM/TO instruction is executed	Stop /Operation continues	OFF/ON	Flashing/ON	SP. UNIT ERROR
	No parameters	On switching on the power or resetting	Stop	OFF	Flashes	MISSING PARA.
	Boot error	On switching on the power or resetting	Stop	OFF	Flashes	BOOT ERROR
	Memory card operation error (Default ... stop) *1	When memory card is inserted/removed	Stop /Operation continues	OFF/ON	Flashing/ON	ICM. OPE ERROR
	File setting error	On switching on the power or resetting	Stop	OFF	Flashes	FILE SET ERROR
	File access error (Default ... stop) *1	When an instruction is executed	Stop /Operation continues	OFF/ON	Flashing/ON	FILE OPE. ERROR
	Cannot execute instruction	On switching on the power or resetting	Stop	OFF	Flashes	CAN'T EXE. PRG.
Parameter errors	Parameter setting error	• On switching on the power or resetting • On switching from STOP to RUN	Stop	OFF	Flashes	PARAMETER ERROR
	Link parameter error	• On switching on the power or resetting • On switching from STOP to RUN	Operation continues	ON	ON	LINK PARA. ERROR
	SFC parameter error	On switching from STOP to RUN	Stop	OFF	Flashes	SFC PARA. ERROR

*1: Can be changed to operation continues by GPPQ parameter setting.
*2: GPPQ parameters can be set so that no check is performed, or so that no check is performed when SM251 is ON.
*3: GPPQ parameters can be set so that no check is performed.

Self-Diagnosis List (Continued)

Diagnosis Details		Diagnosis Timing	CPU Status	LED Statuses		LED Indicator Message (Q3A/Q4ACPU Only)
				RUN	ERROR	
Program error	Instruction code check	• On switching on the power or resetting • On switching from STOP to RUN	Stop	OFF	Flashes	INSTRCT CODE ERR.
	No END instruction	• On switching on the power or resetting • On switching from STOP to RUN	Stop	OFF	Flashes	MISSING END INS.
	Pointer setting error	• On switching on the power or resetting • On switching from STOP to RUN	Stop	OFF	Flashes	CAN' T SET (P)
	Pointer setting error	• On switching on the power or resetting • On switching from STOP to RUN	Stop	OFF	Flashes	CAN' T SET (I)
	Operation check error (Default ... stop) *1	When an instruction is executed	Stop /Operation continues	OFF/ON	Flashing/ON	OPERATION ERROR
	FOR-NEXT instruction configuration error	When an instruction is executed	Stop	OFF	Flashes	FOR-NEXT ERROR
	CALL-RET instruction configuration error	When an instruction is executed	Stop	OFF	Flashes	CAN'T EXECUTE (P)
	Interrupt program error	When an instruction is executed	Stop	OFF	Flashes	CAN'T EXECUTE (I)
	Cannot execute instruction	When an instruction is executed	Stop	OFF	Flashes	INST. FORMAT ERR.
	Extend instruction error (Default ... stop) *1	When an instruction is executed	Stop /Operation continues	OFF/ON	Flashing/ON	EXTEND INST. ERR.
	SFC program configuration error	On switching from STOP to RUN	Stop	OFF	Flashes	SFCP. CODE ERROR
	SFC block configuration error	On switching from STOP to RUN	Stop	OFF	Flashes	CAN'T SET (BL)
	SFC step configuration error	On switching from STOP to RUN	Stop	OFF	Flashes	CAN'T SET (S)
	SFC syntax error	On switching from STOP to RUN	Stop	OFF	Flashes	SFCP. FORMAT ERR.
	SFC operation check error (Default ... stop) *1	When an instruction is executed	Stop /Operation continues	OFF/ON	Flashing/ON	SFCP. OPE. ERROR
	SFC program execution error	On switching from STOP to RUN	Operation continues	ON	ON	SFCP. EXE. ERROR
	SFC block execution error	When an instruction is executed	Stop	OFF	Flashes	BLOCK EXE. ERROR
	SFC step execution error	When an instruction is executed	Stop	OFF	Flashes	STEP EXE. ERROR

Self-Diagnosis List (Continued)

Diagnosis Details		Diagnosis Timing	CPU Status	LED Statuses		LED Indicator Message (Q3A/Q4ACPU Only)
				RUN	ERROR	
CPU error	Watchdog timer error	Constant check	Stop	OFF	Flashes	WDT ERROR
	Program time over	Constant check	Operation continues	ON	ON	PRG. TIME OVER
Annunciator check		When an instruction is executed	Operation continues	ON	OFF	F**** *4
CHK instruction check		When an instruction is executed	Operation continues	ON	OFF	<CHK>ERR ***-*** *5

*1:Can be changed to "operation continues" by GPPQ parameter setting.
*4:The detected annunciator number is displayed at ****.
*5:The detected contact and coil numbers are displayed at ***.

9.3.1 Interruption due to error detection

The QnACPU can execute an interrupt program when an error occurs.
In the case of errors for which operation can be set to continue or stop in PC RAS setting in the parameter mode, this function is only executed when "Resume" is set. If "Pause" is set for the error, a stop error interrupt program (132) is executed.
The relevant errors are shown below.

Interrupt Pointer	Corresponding Error Message
I32	All stop errors
I33	AC DOWN
I34	UNIT VERIFY ERR. FUSE BREAK OFF SP.UNIT ERROR
I35	OPERATION ERROR SFCP OPE.ERROR SFCP EXE.ERROR
I36	ICM.OPE.ERROR FILE OPE.ERROR
I37	EXTEND INS.ERR.
I38	PRG.TIME OVER
I39	CHK instruction Annunciator detection
140 to 147	Vacant

Errors for which the operation mode on occurrence of the error is "Resume", or for which "Resume" is set if it is possible to select "Pause/Resume"

(For details on interrupt pointers, refer to the QnACPU Programming Manual (Fundamentals).)

9.3.2 LED display in the event of an error

In the event of an error, the LED/LED indicator on the front face of the CPU module gives a visual indication.
For details on the operation of the LED and LED indicator, see Section 9.9.

9.3.3 Resetting errors

With the QnACPU, it is only possible to reset errors for which operation is set to continue when that error occurs.

The procedure for resetting an error is as follows.

(1) Eliminate the cause of the error.

(2) Store the error code to be reset in special register SD50.

(3) Turn special register SM50 ON.

(4) The error is reset.

If the CPU is reset with the error reset, the special relay and special register relating to the error, and the LED/LED indicator, return to their state before the error occurred.
If the same error occurs again after the error has been reset, it is recorded in the fault history again.

To reset multiple detected annunciators, only the first detected F number is reset.

POINT

When an error is reset by storing its error code in SD50, the lower two digits of the error code are ignored.
Example:
If errors with error codes 2100 and 2111 have occurred, and error code 2100 is reset, error code 2111 is also reset.

9.4 Fault History

With the QnACPU, the results detected with the self-diagnosis function can be recorded in memory, with the detection time appended, as a fault history.

POINT

Since the internal clock of the QnACPU is used for setting the detection time, always set the correct time before using the CPU.

(1) Storage area

- The latest 16 faults are stored in the fault history storage memory of the CPU, which is latched.
- It is possible to store more than 16 faults by storing the extra ones in files in a memory card by making the appropriate setting in the PC RAS settings in the GPPQ parameter mode.
- If a discrepancy arises between the parameters and memory card fault history due to (a) or (b) below, the contents of the error history files are cleared and the 16-point data of the fault history storage memory of the CPU is transferred to the history file.

(a) The number of fault records in the history file as set in the parameters is changed part way through.

(b) A memory card whose capacity does not match the number of fault records set in the parameters is installed.

- The storage area for the fault history file complies with the following.

Storage area	File in set memory card
Number of fault records storable	Max. 100 (can be changed) *1

*1: When the number of faults that can be stored is exceeded, the oldest fault record is cleared and the newest stored in its place.

POINT

Even if the fault history file set in the parameters does not exist in the memory card, no CPU error occurs.
The CPU performs only the processing to store faults that occur in the fault history storage file.

(2) Clearing the fault history

The fault history is cleared by using the fault history clear function in the PC menu in the PC diagnosis mode of GPPQ.

9.5 System Protect

The QnACPU features a number of functions that protect against program changes ("system protect") by restricting general data processing (access processing from GPPQ, serial communication modules, etc.) by third parties other than designers.
The following system protect functions are available.

Object of Protection	Files for Which Protection is Effective	Details of Protection	Method	When Effective	Remark
CPU as a whole	All files	Batch prohibition of writing/control directions with respect to the CPU.	Set system setting switch SW5 of the CPU to ON. (See Section 15.2)	Constant check	Effective for devices too
Memory card units	All files	Establishes write protect for the memory card and prohibits writing.	Set the memory card's write protect switch to ON. (See Section 18.2)	Constant check	
Drive units	Parameter program	Registers entry codes for the following settings in relation to a specific drive (for example the internal memory): (1) Prohibiting read/write display (2) Prohibiting writing	Perform password registration. (See Section 9.6)	Constant check	
File units	All files	Changes attributes file by file in the following ways: (1) Prohibiting read/write display (2) Prohibiting writing	Change file attributes by password registration. (See Section 9.6)	Constant check	

"Control direction", "read/write display" and "writing" in the table above have the following meanings:

Item	Description
Control direction	Direction to control CPU operation remotely (remote RUN, remote STOP, etc.)
Read/write display	Operations such as program reading and writing.
Writing	Operations that involve write processing, such as program writing and testing.

9.6 Password Registration

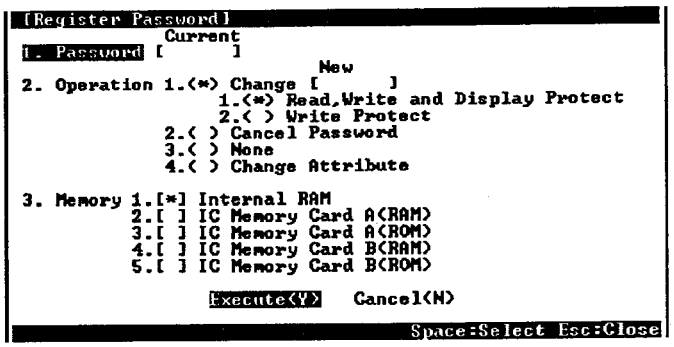
Passwords serve to prohibit reading and overwriting of data such as programs, comments, etc., in the QnACPU from a peripheral device.
In password registration, the parameter files and program files of a designated memory (internal memory, memory card) are made the object of the entry code. There are two types of registration, as follows:

- File names are not displayed, and reading and writing is prohibited.
- File writing is prohibited. (Reading is possible).

When an password is registered, file operations from a peripheral device are not possible without inputting the entry code registered in the CPU.

(1) Registering an password

Entry codes are registered using the entry code registration function in the PC menu in the online mode of GPPQ.



The following is an explanation of each item in the screen.

(a) "Password" ..... If an password has been registered in the CPU, input the registered password to make file operations possible.
If an incorrect password is input, file operations will not be possible.

(b) "Operation" ..... 1) Change : Register a new password in the CPU.
Or, if the password matches, change the password.

(a) Read, Write and Display Protect : File names in the designated memory cannot be displayed or written to.

(b) Write Protect : Files in the designated memory cannot be written to. Reading is possible.

2) Cancel Password : If the password matches, the registered password is deleted from the CPU.

3) None : The present password is recorded at the GPPQ only and is not registered at the CPU.

4) Change Attribute : Allows file read/write display or writing to be prohibited in file units. (Operation possible even if no entry code is registered.)

(c) "Memory" ..... Designate the memory for which the password is to be registered.

POINTS

(1) Password registration is valid for parameter files and program files only.
It is not valid for other types of file. Other types of file can be protected by changing attributes for each file.

(2) The password registered in the CPU cannot be read from the CPU. If you forget the password, CPU file operations will be impossible. Keep a record of the password, e.g. on paper, and store it in a safe place.

9.7 Online I/O Module Change

Normally, when an I/O module is mounted or removed while the power to the QnACPU is ON, a "UNIT VERIFY ERROR" occurs and PC CPU operation stops or control is executed with I/O numbers that are different from the set numbers.
The online I/O module replacement function allows an I/O module to be replaced while the power to the PC CPU is ON without causing a "UNIT VERIFY ERROR".

POINT

The online I/O module replacement function can only be used with I/O modules.
It cannot be used with special function modules.

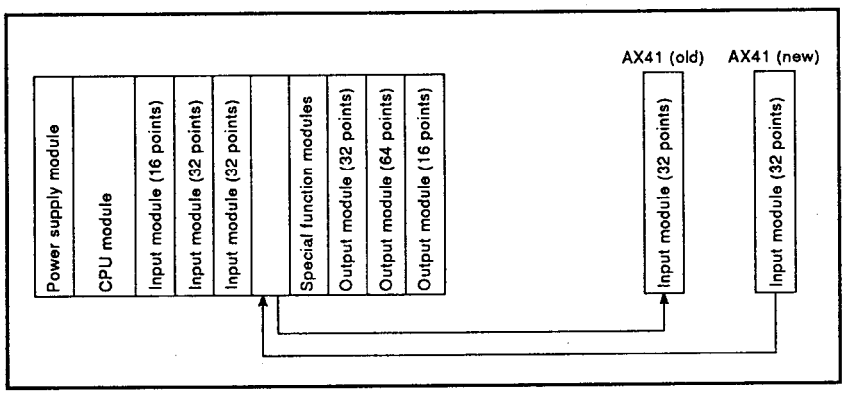


Fig. 9.2 Online I/O Module Replacement

(1) Application

The online I/O module replacement function is used to replace an I/O module at which an error has occurred while continuing control by the sequence program.

(2) Replacement method

The following two methods can be used to replace an I/O module while online.

(a) Use "Connect Unit" in the PC menu in the GPPQ PC diagnosis mode.

(b) Use special relays/special registers.

(3) Procedure when using GPPQ

(a) Designate the I/O module to be replaced with "Connect Unit" in the PC menu in the PC diagnosis mode.

(b) Replace the designated module.

(4) Procedure when using special relays/special registers

(a) Use the following procedure when using special relays/special registers:

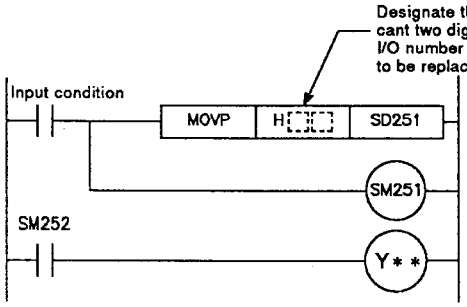
1) Set the most significant two digits of the three-digit expression for the head I/O number of the I/O module to be replaced in SD251 (replaced I/O module head I/O No. storage register).
Examples: Head I/O No. = 070 → set number is H07
Head I/O No. = 170 → set number is H17

2) Turn SM251 (I/O replacement flag) ON.

3) After confirming that SM252 (replacement enabled flag) has come ON, replace the designated I/O module.

4) Turn SM251 OFF. (SM252 turns OFF automatically.)

(b) Setting data in SD251 and turning SM251 ON/OFF can be done either by using the sequence program or a peripheral device.

(a) Method when Using a Sequence Program	(b) Method when Using GPPQ
A sequence program of the type shown below can be used to replace a designated I/O module while the QnACPU is in the "RUN" status.  *The sequence program shown above can be written from GPPQ in the RUN state. **Procedure** (1) Turn the input condition ON, set the number of the I/O module to be replaced in SD251 and turn SM251 ON. (2) After confirming that output Y** has come ON, replace the designated I/O module. (3) Turn the input condition OFF and turn SM251 OFF. (This starts operation of the module after replacement.)	A designated I/O module can be replaced in the GPPQ test mode. The QnACPU can be in the RUN, STOP, or PAUSE state. **Procedure** (1) Connect a peripheral device capable of GPP functions to the QnACPU. (2) In the test mode, set the most significant two digits of the head I/O number of the module to be replaced in SD251. (3) In the test mode, set (turn ON) SM251. (4) In the monitor mode, confirm that SM252 has come ON, then replace the designated I/O module. (5) In the test mode, reset (turn OFF) SM251. (This starts operation of the module after replacement.)

9.8 System Display

The following items can be checked on connecting a peripheral device capable of GPP functions to the QnACPU:

(1) The following information relating to the modules actually mounted on the base unit:

(a) Type
(b) Number of occupied I/O points
(c) Head X/Y number

(2) The following module information set in the parameters:

(a) Type
(b) Number of occupied I/O points
(c) Model Name

(3) The following information relating to the CPU:

(a) Status of the RUN/STOP key switch
(b) Status of the system setting switches
(c) LED statuses

These items can be checked using the detail HELP display and CPU panel items in the display menu of the GPPQ PC diagnosis mode.

9.9 LED/LED Indicator Indications

The QnACPU module has LEDs on its front face that indicate the operating state of the CPU. In addition, Q3ACPU and Q4ACPU feature a LED display.
The meanings of the LED and LED indicator indications are explained below.

9.9.1 LED

(1) The meanings of the indications of each of the LEDs are given in the table below.

LED Name	Indication Details
RUN	Indicates the operating state of the CPU. ON : A sequence program is being executed with the RUN/STOP key switch in the RUN or STEP RUN position. OFF : Operation is stopped, with the RUN/STOP key switch in the STOP, PAUSE, or STEP RUN position. Or, an error that stops operation has been detected. Flashing: The key switch has been turned from STOP to RUN after writing a program in the STOP state. To light the RUN LED, either turn the key switch from RUN to STOP and then back to RUN, or reset operation using the key switch.
ERROR	Indicates the CPU error detection status. ON : A self-diagnosis error that does not stop operation, other than a battery error, has been detected. (Errors for which the operation mode when the error occurs has been set to "Resume" in PC RAS setting in the parameter mode.) OFF : Normal Flashing: An error that stops operation has been detected.
USER	Indicates the CHK instruction detection status, and annunciator (F) statuses. ON : An error has been detected by means of the CHK instruction, or an annunciator, F, has come ON. OFF : Normal Flashing: When latch clear is performed.
BAT. ALARM	Indicates the battery statuses of the CPU itself and the memory card. ON : A battery error has occurred due to low battery voltage. OFF : Normal
BOOT	Indicates the boot operation execution status. ON : Execution is completed. OFF : The boot operation has not been executed.

(2) An LED that is currently ON can be turned OFF by using the following operations (other than the reset operation).

Method for Turning the LED OFF	LED Name			
	ERROR	USER	BAT. ALARM	BOOT
Eliminate the cause of the error, then execute the LEDR instruction.	o	o	o	x
Eliminate the cause of the error, then press the INDICATOR RESET button.	o	o	o	x
Eliminate the cause of the error, then reset the error using special relay SM50 and special register SD50. (Restricted to error which do not stop operation.)	o	o	o	x
Turn the LED OFF by using special relay SM202 and special register SD202.	x	o	x	o

o : Valid x : Not valid

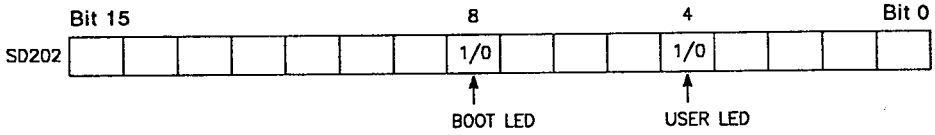
*1 Explanation of special relays and special registers

SM50 ..... On turning from OFF to ON, resets the error corresponding to the error code stored in SD50.

SD50 ..... Stores the error code of the error to be reset. (For details on error codes, see Section 20.3.2.)

SM202 .... On turning from OFF to ON, turns OFF the LEDs corresponding to each of the bits of SD202.

SD202 .... Designates the LED to be turned OFF (the LEDs that can be turned OFF are the USER LED and the BOOT LED only).



A bit setting of "1" indicates that the bit is to be turned OFF, "0" indicates that it is not to be turned OFF.
The setting possibilities are indicated below (all hexadecimal notation):

- To turn both LEDs OFF: SD202 = 110H
- To turn only the BOOT LED OFF: SD202 = 100H
- To turn only the USER LED OFF: SD202 = 10H

(3) Method for stopping ERROR LED, USER LED, and BAT.ALARM LED indications

The ERROR LED, USER LED and BAT.ALARM LED conform to the same order of priority as described for the LED indicator in Section 9.9.2. If an error item number is deleted from this order of priority, the LED will not light even if the error corresponding to that error item number occurs. (For details on the setting method, refer to the POINT in Section 9.9.2.)

9.9.2 LED indicator

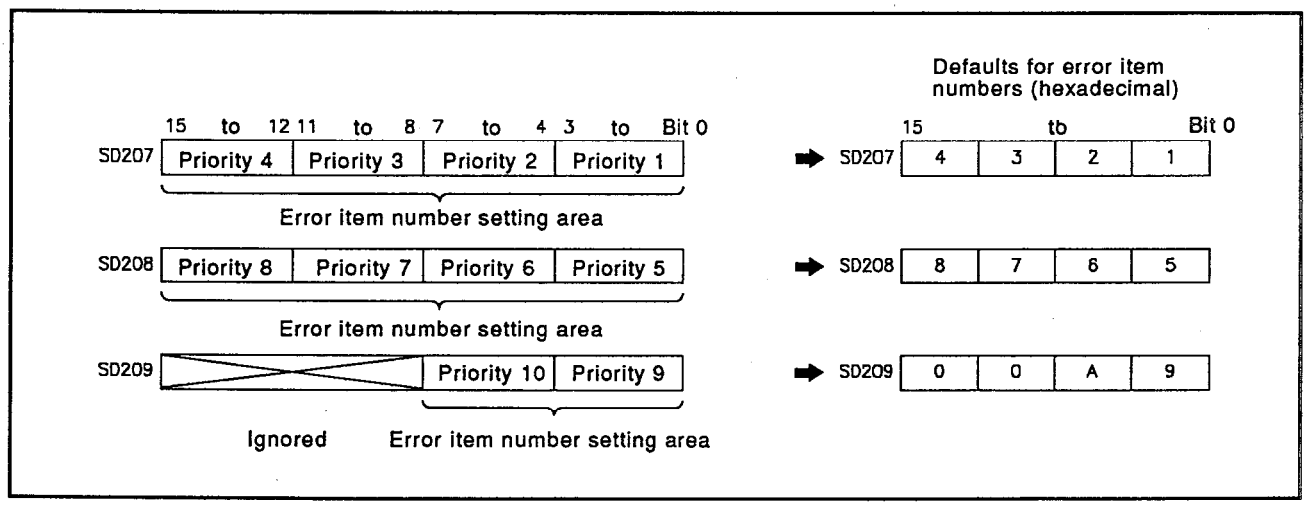
In addition to LEDs, the Q3ACPU and Q4ACPU are provided with an LED indicator on the front face of the CPU module. This indicator displays the following items:

1) Error messages
2) CHK instruction numbers
3) Annunciator numbers, annunciator comments, the time when an annunciator was switched ON, etc.
4) Character string and comment displays in accordance with LED instructions
5) Time display

(1) If reasons for more than one indication exist at the same time, the display conforms to the following conditions.

1) Errors that stop operation are displayed unconditionally.
2) Errors that do not stop operation are displayed in accordance with error item numbers in an order of priority set by default.
The order of priority can be changed. (The settings are made with special registers SD207 to SD209).
3) If more than one error with the same priority occurs, the one that was detected first is displayed.

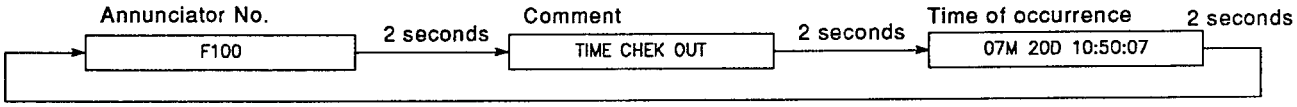
The order of priority is set in special registers SD207 to SD209 as shown below.



The defaults for the error item numbers set in special registers SD207 to SD209 and order of priority are given in the table below.

Order of Priority	Error Item No. (Hex.)	Description	Remark
1	1	AC DOWN	Power supply cut
2	2	UNIT VERIFY ERR. FUSE BREAK OFF SP. UNIT ERROR	I/O module verify error Blown fuse Special function module verify error
3	3	OPERATIN ERROR LINK PARA. ERROR SFCP OPE. ERROR SFCP EXE. ERROR	Operation error Link parameter error SFC instruction operation error SFC program execution error
4	4	ICM. OPE. ERROR FILE OPE. ERROR EXTEND INST. ERROR	Memory card operation error File access error Extend instruction error
5	5	PRG. TIME OVER	Constant scan setting time over error Low-speed execution monitoring time over error
6	6	CHK instruction	
7	7	Annunciators	
8	8	LED instruction *2	
9	9	BATTERY ERR.	
10	A	Clock data	

*2 For annunciators, it is possible to set alternating display of comment and time of occurrence by parameter setting (PC RAS setting).
Example: When the setting is made — in PC RAS setting in the parameter mode — to display comments and time of occurrence:

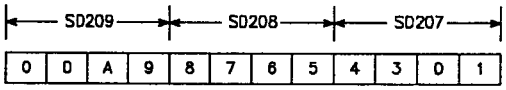


However, the time of occurrence is displayed for the first annunciator only. The time display for the second and subsequent annunciators is as follows: "--M--D--:--:--".

POINTS

(1) To set the LED and LED indicator to remain OFF even when one of the errors indicated above occurs, set "0" in the error item number setting area set in SD207 to SD209.

Example: To set the ERROR LED to remain OFF and the LED indicator to remain blank when a fuse blown error occurs, set "0" in the item number setting area whose item number is "2".



Since the item number "2" is not set, the ERROR LED will remain OFF and the LED indicator blank if a fuse blown error is detected. The ERROR LED will also remain OFF and the LED indicator blank if another error whose error item number is "2" is detected (I/O module verify error, special function module verify error).

(2) Even if the LED is set to remain OFF and the LED indicator blank, SM0 (the diagnosis error flag) is still turned ON, SM1 (the self-diagnosis error flag) is still turned ON, and the error code is stored in SD0 (CPU diagnosis error register).

(2) A displayed message can be cleared by the same procedure as used to turn an LED OFF.
(See Section 9.9.1 (2)).
However, if several display causes have occurred at the same time, the next message will be displayed when the currently displayed message is cleared.

10. OTHER FUNCTIONS

10.1 Function List

The functions other functions are listed in the table below.

Item	Description	Refer to
Constant scan	Function that executes a program at constant time intervals regardless of the actual program scan time.	Section 10.2
Latch function	Function that retains the data in devices when the power is switched off and on resetting.	Section 10.3
Setting for output status on switching from STOP to RUN	Function that sets the output (Y) status when the CPU is switched from STOP to RUN (re-output of the output before STOP, or output after operation execution).	Section 10.4
Clock function	Function that runs the internal clock of the CPU.	Section 10.5
Remote operation	Functions for operating the QnACPU from a remote location.	Section 10.6
Remote RUN/STOP	Function to start and stop CPU operation.	Section 10.6.1
Remote STEP-RUN	Function to execute step by step CPU operation.	Section 10.6.2
Remote PAUSE	Function to temporarily stop the CPU.	Section 10.6.3
Remote RESET	Function to reset the CPU.	Section 10.6.4
Remote latch clear	Function to clear the CPU latch data.	Section 10.6.5
Relationship between remote operation and CPU key switch	Explains the relationship between the CPU key switch setting and operation when performing remote operation.	Section 10.6.6
Terminal setting	Functions using the Q6PU programming unit's display and key input.	Section 10.7
Message display	Function to display messages on the display of the Q6PU.	Section 10.7.1
Key input operation	Function to read key input from the Q6PU.	Section 10.7.2
Reading module access time intervals	Function for monitoring the access time intervals (time between acceptance of one CPU access and acceptance of the next CPU access) for special function modules, network modules, and peripheral devices.	Section 10.8

For details on operations at GPPQ, refer to the SW□IVD-GPPQ Operating Manual (Online).
For details on operations at a Q6PU, refer to the Q6PU Operating Manual.

10.2 Constant Scan

(1) What is constant scan?

Normally, the scan time of the QnACPU varies because the processing time is not constant; it differs according to the execution or non-execution of the instructions used in the sequence program.
Constant scan is a function whereby the sequence program is repeatedly executed while maintaining a constant scan time.

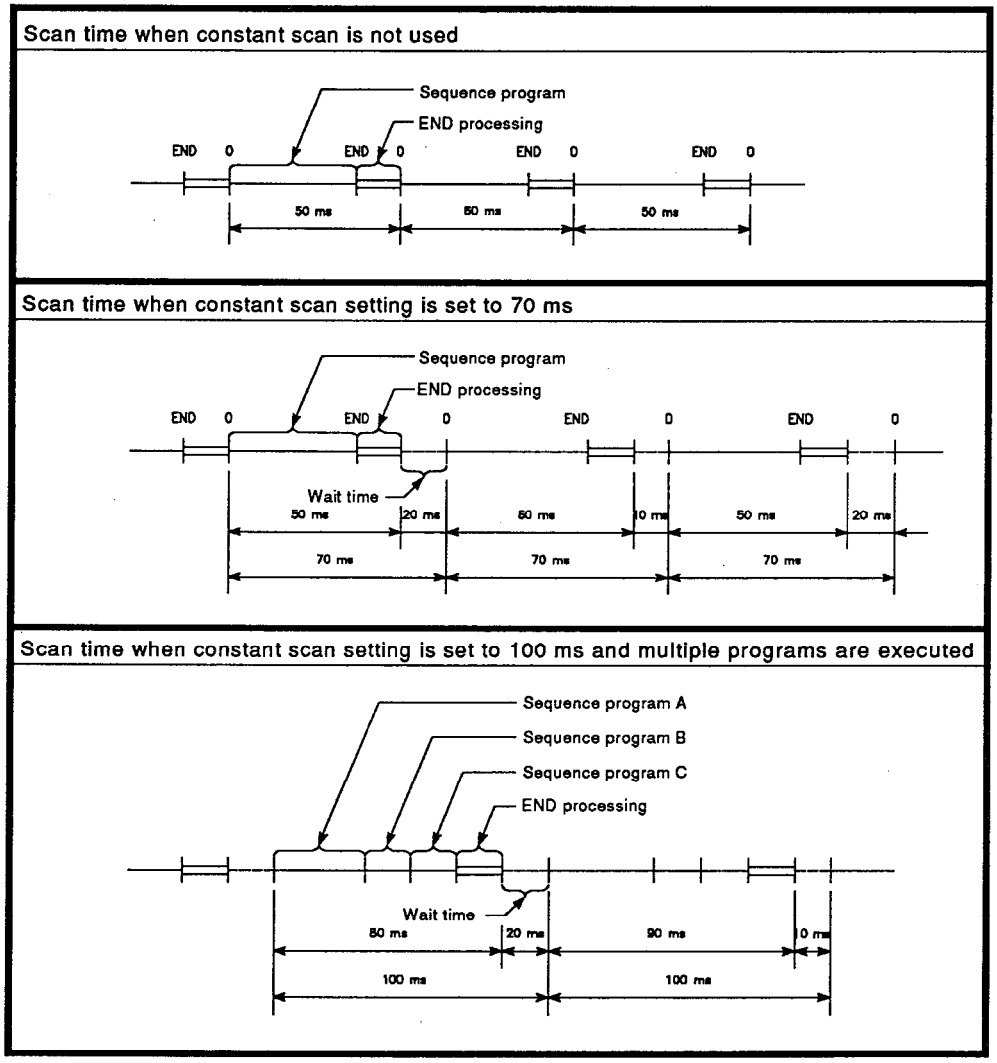


Fig. 10.1 Operation for Constant Scan Time

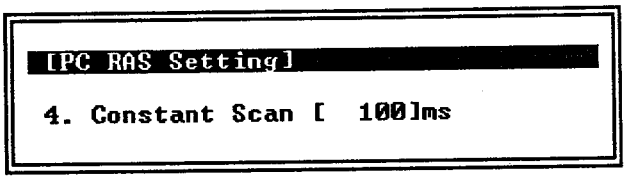
When low-speed execution type programs are used, either this constant scan function or a low-speed execution time must be set.
(For details, refer to the QnACPU Programming Manual (Fundamentals).)

(2) Setting the constant scan time

(a) The constant scan time is set in PC RAS setting in the GPPQ parameter mode.

- If the constant scan function is to be used, set a constant scan time.
- If not using the constant scan function, set a blank for the constant scan time.

Example: Setting a constant scan time of 100 ms



(b) Set a constant scan time that is longer than the maximum scan time of the sequence program. If the scan time of the sequence program is longer than the set value for constant scan time, the QnACPU detects an error (SD0 = 5010) and the sequence program is executed in accordance with its own scan time, ignoring the constant scan time setting.
Also, make sure that the constant scan time setting is shorter than the set time for WDT. If it is longer than the set time for WDT, the QnACPU detects a WDT error and program execution is stopped.

Set the constant scan time within the following range.

Set time for WDT > Set time for constant scan > Maximum scan time of sequence program

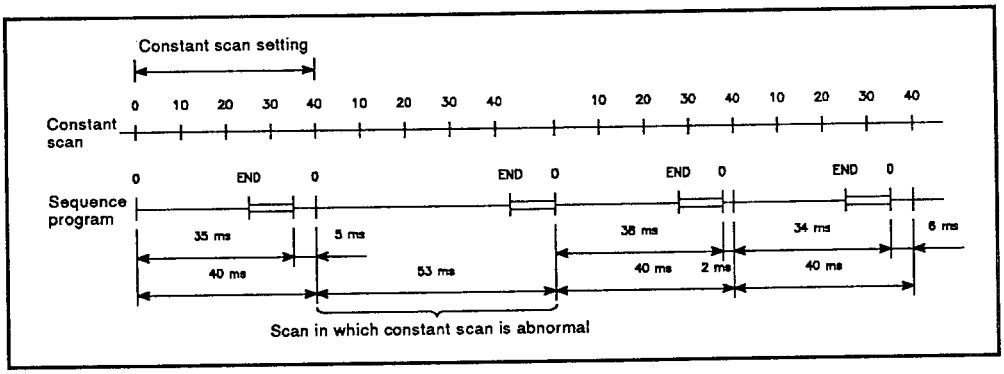


Fig. 10.2 Operation When Scan Time is Longer Than Constant Scan Time Setting

(c) Sequence program processing is suspended in the wait time in the interval between END processing of the sequence program and the start of the next scan.
However, if an interrupt cause occurs, or if there is a low-speed execution program, the interrupt program or low-speed execution program is executed after execution of END processing.

(d) Constant scan time error

If there is a low-speed execution type program when the constant scan function is being used, the constant scan time may be shifted by the time shown below:

$$\text{Error} = \begin{array}{l}\text{Maximum processing time}\\ \text{of one instruction in the}\\ \text{low-speed execution program}\end{array} + \underline{\text{low-speed END processing time}}$$

(Time taken to execute the END processing for a low-speed execution program)

Low-speed execution type programs are executed part by part in the surplus time after the constant scan time. If the end of the constant scan time falls part way through execution of an instruction with a long processing time, processing of the instruction is completed before ending the scan. This time extension to complete execution of the instruction is the constant scan error.
For details on instruction processing times, refer to the QnACPU Programming Manual (Common Instructions).

10.3 Latch Function

When the QnACPU power is turned ON, the CPU is reset using the RUN/STOP key switch, or a power interruption lasting longer than the allowable momentary power interruption time occurs, the values in the various devices of the QnACPU are cleared and the default values are set in the devices (bit devices: OFF, word devices: 0).
The latch function serves to retain the data in the devices when the QnACPU power is turned ON, the CPU is reset with the RUN/STOP key switch, or a power interruption longer than the allowable momentary power interruption time occurs.
The operations in the program are the same whether or not the latch function is used.

(1) Applications of the latch function

The latch function can be used when carrying out continuous control, to retain data such as production quantities, numbers of defects, addresses, etc., even if a power interruption longer than the allowable time occurs.

(2) Devices that can be latched

(a) The following devices can be latched:

1) Latch relays
2) Link relays
3) Annunciators
4) Edge relays
5) Timers
6) Retentive timers
7) Counters
8) Data registers
9) Link registers

POINT

Even if a latch designation is set for a device, the device will not be latched if a local device designation or device initial value designation is made.

(b) The latch range is set in device setting in the GPPQ parameter mode. In latch range setting, it is possible to set ranges within which the latch clear key is effective and ranges within which it is not effective.
For details on the latch range for each device, refer to the QnACPU Programming Manual (Fundamentals).

POINTS

The contents of devices in the latch range are retained by the battery (A6BAT) installed in the CPU module.

(1) Even if a sequence program is stored in a ROM in a memory card and ROM operation is used, a battery is required for the latch function.

(2) If the battery connector is disconnected from the CPU module connector while the CPU module power supply is OFF, the data in the devices in the latch range is lost.

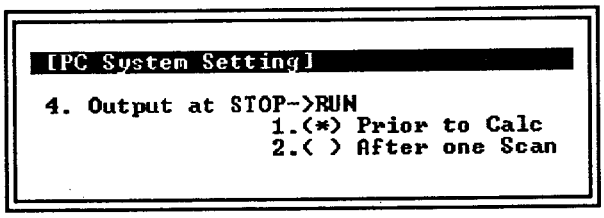
(3) Clearing device data in the latch range

(a) To clear the data in the devices in the latch range and set the default values instead, execute the "latch clear" operation. When latch clear is executed, the data in devices in the non-latched range is also cleared.
However, the devices for which the latch clear key has been set as ineffective in the parameters are not cleared on executing latch clear.

(b) The method for executing latch clear is described in Section 12.4.

10.4 Setting of the Output (Y) Status When Switching from STOP to RUN

When the operating state is switched from RUN to STOP, the outputs (Y) in the RUN state are stored in the PC CPU.
This function determines whether, on switching from STOP back to RUN, these stored outputs are re-output or outputs are made after sequence program operation; it is set in PC system setting in the GPPQ parameter mode.



(a) Re-output (Prior to Calc) ..... The output (Y) statuses immediately before the STOP state was established are output, then sequence program operation is executed.

(b) Output after operation execution... (After one Scan) All outputs (Y) are cleared, and sequence program operation is executed, before outputs (Y) are output.

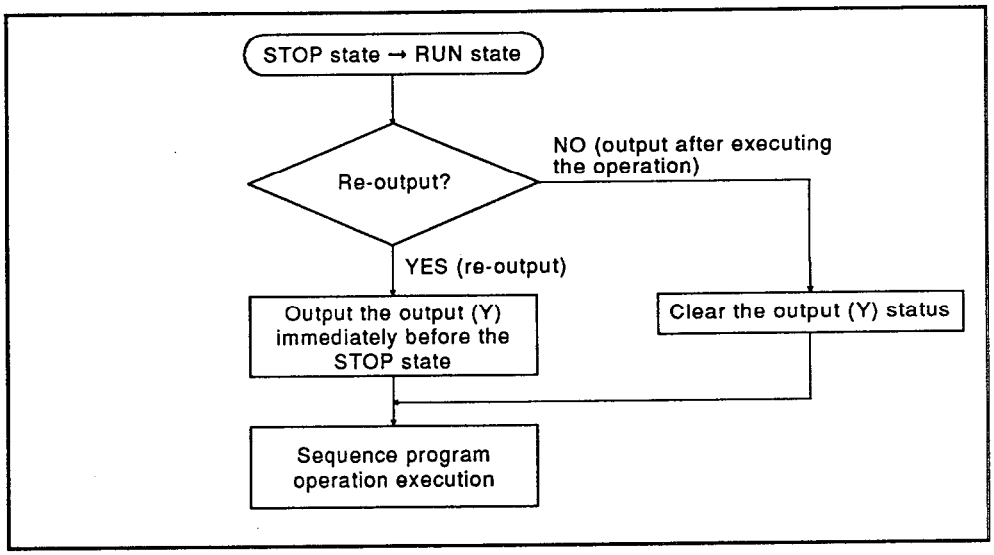


Fig. 10.3 Processing When the Operating State is Switched from STOP to RUN

10.5 Clock Function

With QnACPU, the CPU module has a clock function.
Since clock data can be read by a sequence program using the clock function, this data can be used for time management.
In addition, the clock data can also be used for time management of functions executed by the CPU system, such as the fault history.
Clock operation by the clock function is continued by means of the battery when the PC power is turned OFF or a power interruption longer than the allowable momentary power interruption time occurs.

POINT

Since the clock data is used by the CPU system — e.g. for the fault history — make sure that the time is set accurately when using the CPU for the first time.

(1) Clock data

The clock data is the year, month, day, hour, minute, second, and day of the week data used by the clock devices in the PC CPU, as tabled below.

Data Name	Description	
Year	Last two digits of year	
Month	1 to 12	
Day	1 to 31 (automatic recognition of leap years)	
Hour	0 to 23 (24-hour system)	
Minute	0 to 59	
Second	0 to 59	
Day of the week	0	Sunday
	1	Monday
	2	Tuesday
	3	Wednesday
	4	Thursday
	5	Friday
	6	Saturday

(2) Accuracy

The accuracy of the clock function depends on the ambient temperature, as shown below.

Ambient Temperature (°C)	Accuracy (Error in Seconds Per Day)
0	–2.3 to +4.4 (TYP.+1.8)
+25	–1.1 to +4.4 (TYP.+2.2)
+55	–9.6 to +2.7 (TYP.–2.4)

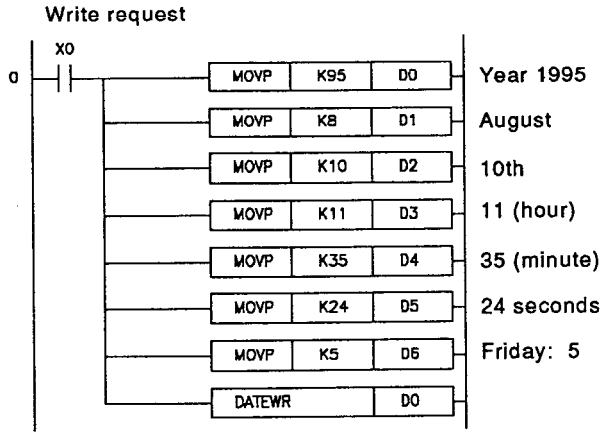
(3) Writing clock data to the clock devices

(a) Use the following procedure to write clock data to the clock devices.

1) Writing from a peripheral device

- When using GPPQ, clock data can be written to the clock devices by using the clock setting option in the PC menu of the PC diagnosis mode.
- When using Q6PU, clock data can be written to the clock devices by using the clock monitor option in the monitor functions under the PC system menu in the other mode.
  (For details on the operations at a particular peripheral device, refer to the Operating Manual for that device.)

2) Writing from a program
Clock data is written to the clock devices by using a clock instruction (DATEWR).
An example of a program that writes clock data by using the clock data write instruction (DATEWR) is shown below.



For details on the DATEWR instruction, refer to the QnACPU Programming Manual (Common Instructions).

POINTS

(1) Clock data is not written into the clock devices on delivery.
Write clock data to the clock devices before using the QnACPU.

(2) Even if only changing part of the clock data, write data for all the clock data settings to the clock devices again.

(3) If data that cannot be an actual clock setting is written to the clock devices, normal clock operation will not be possible.

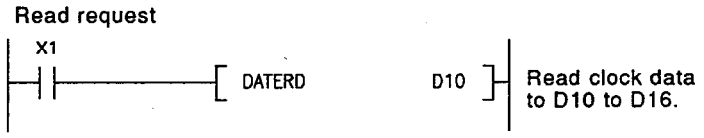
Example:

Setting of "13" for the month.
Setting of "32" for the date.

(4) Reading clock data

(a) To read clock data to data registers, use a clock data read instruction (DATERD) in the program.

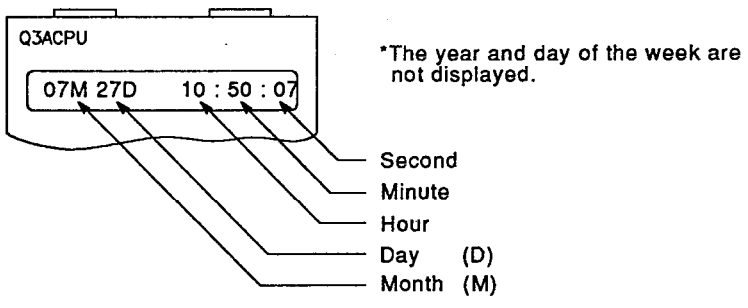
An example of a program using this instruction is shown below.



For details on the DATERD instruction, refer to the QnACPU Programming Manual (Common Instructions).

(b) To read the clock data to SD210 to SD213, turn SM213 ON from a sequence program or a peripheral device.

(c) In the case of CPU modules with a 16-character LED indicator on their front panel, the clock data (month, day, hour, minute, second) can be displayed on the LED indicator.
To display the clock data on the LED indicator, turn SM212 ON.
The clock data display has the lowest priority of any LED display and it will therefore not be displayed if an error occurs or other information has to be displayed for any reason.



(5) Special relays and special registers for reading and writing clock data

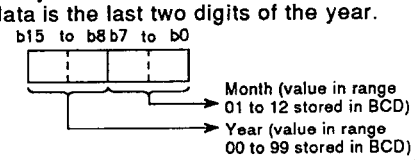
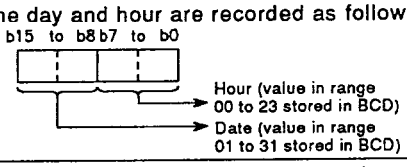
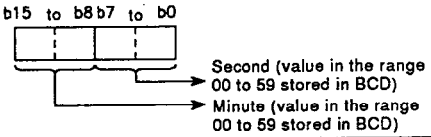
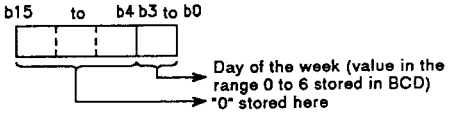
The special relays and special registers used for setting data and reading clock data for clock operation are described here.

(a) Special relays used for the clock function

Device	Name	Description
SM210	Clock data set request	• Used to write clock data to the special registers (SD210 to SD213) and execute clock operation. • Writes the clock data stored in SD210 to SD213 to the clock devices after execution of END processing in the scan in which SM210 turns from OFF to ON.
SM211	Clock data error	• Serves to determine whether or not there are any errors when the clock data is set. • Comes ON if any data is not a BCD record.
SM212 *1	Clock data display	• Displays clock data on the LED indicator on the front panel of the PC CPU. • When SM212 is ON, the clock data is displayed on the LED indicator on the front panel of the CPU module.
SM213	Clock data read request	• Used to read the clock data stored in special registers SD210 to SD213. • When SM213 is ON, the clock data is read to SD210 to SD213 after execution of END processing.

*1: Indicates can only be used with Q3ACPU and Q4ACPU.

(b) Special registers used with clock data

Device	Name	Description
SD210	Clock data (year, month)	• The year and month are recorded as follows. The year data is the last two digits of the year. 
SD211	Clock data (day, hour)	• The day and hour are recorded as follows. 
SD212	Clock data (minute, second)	• The minute and second are recorded as follows. 
SD213	Clock data (day of the week)	• The day of the week is recorded as follows.  • The settings for the day of the week are as follows:

Day of the week	Sun	Mon	Tue	Wed	Thu	Fri	Sat
Stored data	0	1	2	3	4	5	6

10.6 Remote Operation

With QnACPU, the operating status of the CPU can be control from an external source (GPPQ, intelligent special function module, remote contact, etc.).

REMARK

In this section, a serial communication module is used as an example of an intelligent special function module.

10.6.1 Remote RUN/STOP

Remote RUN/STOP means setting the QnACPU to the RUN or STOP state from an external source while the CPU RUN/STOP key switch is set to the RUN position.

(1) Application of remote RUN/STOP

It is convenient to use remote operation to execute remote RUN/STOP in the following cases:

(a) When the CPU is installed in an inaccessible location;

(b) To set a CPU in a control panel to RUN/STOP from an external source.

(2) Operation for remote RUN/STOP

The program operation when remote RUN/STOP is executed is as follows:

(a) Remote STOP .... The program is executed up to the END instruction, then the STOP state is established.

(b) Remote RUN ..... When remote RUN is executed with the CPU in the STOP state due to a remote STOP, the program is executed from step 0.

(3) Method for executing remote RUN/STOP

There are two methods for executing remote RUN/STOP, as follows:

(a) Method using a remote RUN contact

The remote RUN contact is set in PC system setting in the GPPQ parameter mode.

The device range in which the setting can be made is inputs X0 to 1FFF.

Remote RUN/STOP can be executed by switching the remote RUN contact ON and OFF.

1) When the remote RUN contact is OFF, the CPU is in the RUN state.

2) When the remote RUN contact is ON, the CPU is in the STOP state.

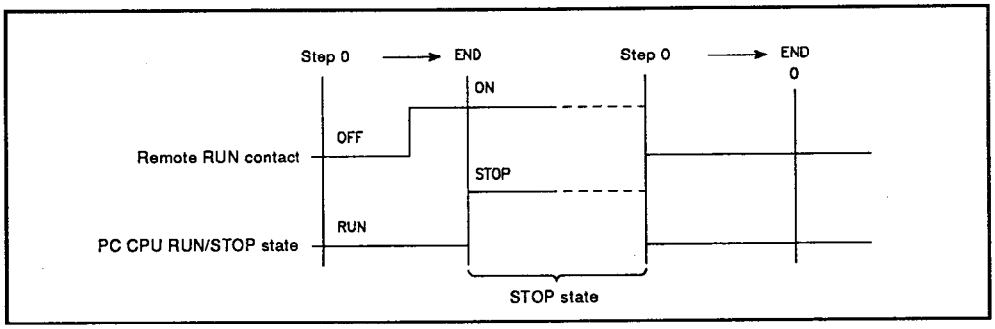


Fig. 10.4 Time Chart for RUN/STOP Switching with Remote RUN Contact

(b) Method using GPPQ, serial communication module, etc.

The CPU can be set to RUN or STOP by remote RUN/STOP operation from GPPQ, or a serial communication module, etc.
For operation from GPPQ, remote operation is performed under the PC menu in any mode.

RUN/STOP control using a serial communication module is achieved by using commands in the dedicated protocol.
For details on serial communication module control, refer to the serial communication module User's Manual.

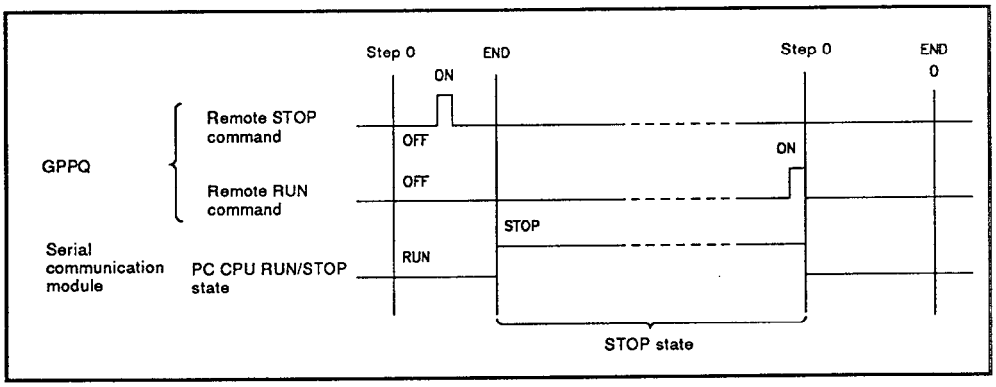


Fig. 10.5 Time Chart for Remote RUN/STOP Switching with GPPQ or Serial Communication Module

(4) Cautions

(a) Since the QnACPU has STOP priority, the following points must be born in mind.

1) If a remote stop is executed with respect to the QnACPU from any one source (remote RUN contact, GPPQ, serial communication module, etc.) the STOP state will be established.

2) In order to set the QnACPU back to the RUN state after it has been set to STOP by a remote STOP, all external sources at which remote STOP is set (remote RUN contact, GPPQ, serial communication module, etc.) must be set to RUN.

REMARK

The RUN and STOP states are defined as follows:

- RUN state..... Status in which the sequence program is repeatedly executed from step 0 to the END instruction.
- STOP state..... Status in which sequence program operation is stopped and all outputs (Y) are OFF.

10.6.2 Remote STEP-RUN

Remote STEP-RUN is a function whereby QnACPU step run operation is executed from GPPQ while the CPU's RUN/STOP key switch is left at the RUN position.
"Step run" means executing program operation one step at a time, starting from the designated step.
For details on step run, see Section 8.7.

(1) Application of remote STEP-RUN

When debugging the system, for example, the program can be executed while checking its execution and the contents of each device.

(2) Method for executing remote STEP-RUN

Use the following procedure to execute remote STEP-RUN.

1) Set the RUN/STOP key switch of the CPU to "RUN".
2) Execute the remote STEP-RUN operation at GPPQ.

10.6.3 Remote PAUSE

The remote PAUSE function establishes the PAUSE function at the QnACPU from an external source while the RUN/STOP key switch of the CPU is set to the RUN position.
The PAUSE function stops CPU operation while retaining the ON/OFF statuses of all outputs (Y).

(1) Application of remote PAUSE

Used, for example in process control, to hold outputs (Y) ON even when the CPU is set to STOP.

(2) Method for remote PAUSE

There are two methods for remote PAUSE, as follows:

(a) Method using a remote PAUSE contact

The remote PAUSE contact is set in PC system setting in the GPPQ parameter mode.

The device range in which the setting can be made is inputs X0 to 1FFF.

1) When END processing is executed for a scan in which the remote PAUSE contact and pause enabled flag (SM206) are both ON, the PAUSE state contact (SM204) comes ON.
After execution up to the END instruction of the scan following the scan in which the PAUSE state contact came ON, the PAUSE state is established and operation is stopped.

2) On turning the remote PAUSE contact OFF or turning SM206 OFF at the GPPQ, the PAUSE state is reset and sequence program operation is executed from step 0 again.

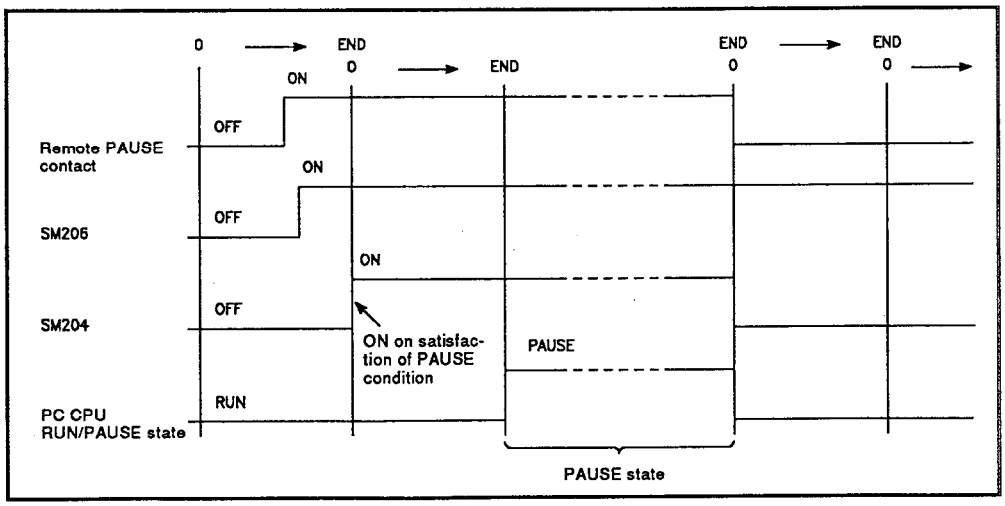


Fig. 10.6 Timing Chart for PAUSE Caused by Remote PAUSE Contact

(b) Method using GPPQ, serial communication module

Remote PAUSE operations can be performed from GPPQ or from a serial communication module.
For operation from GPPQ, remote operation is performed under the PC menu in any mode.

RUN/STOP control using a serial communication module is achieved by using commands in the dedicated protocol.
For details on serial communication module control, refer to the serial communication module User's Manual.

1) On execution of the END processing in the scan in which the remote PAUSE command was received from GPPQ, the PAUSE state contact (SM204) turns ON.
After execution up to the END instruction of the scan following the scan in which the PAUSE state contact came ON, the PAUSE state is established and operation is stopped.

2) When the remote RUN command is received from GPPQ, execution of sequence program operation restarts from step 0.

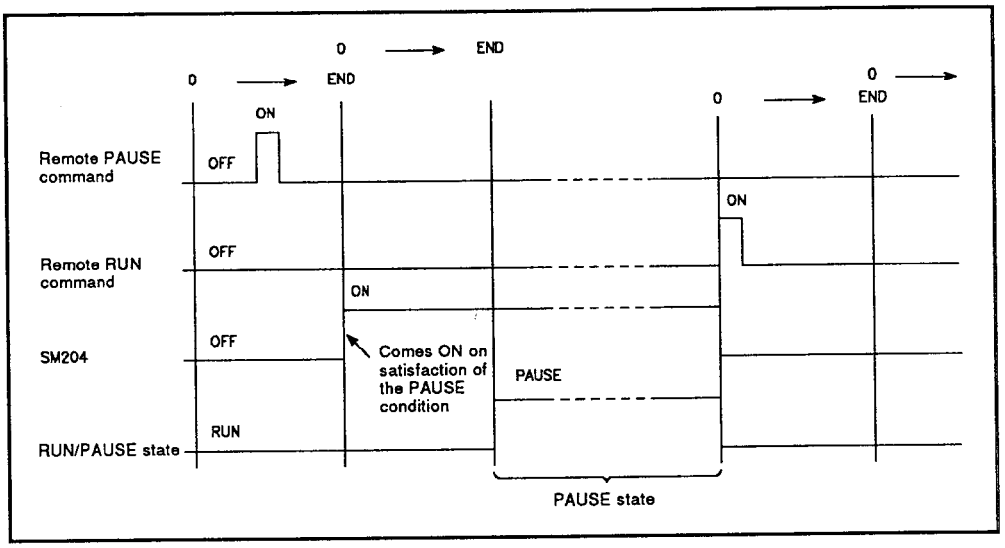
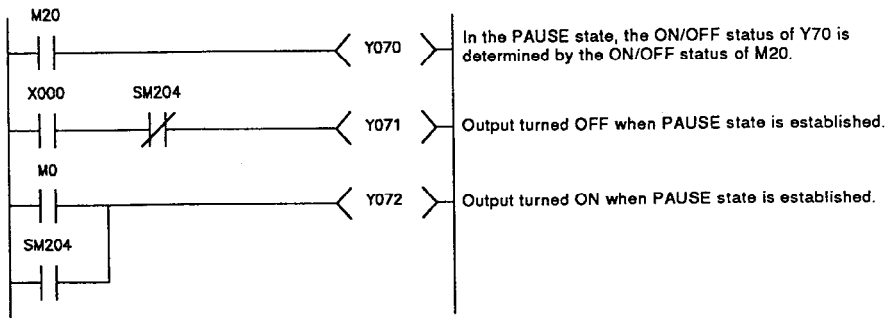


Fig. 10.7 Timing Chart for PAUSE from GPPQ

POINT

To arrange for outputs (Y) to be set ON or OFF when the PAUSE state is established, establish an interlock using the PAUSE state contact (SM204).



10.6.4 Remote RESET

Remote RESET is a function for resetting the QnACPU by operation from an external device while the CPU is in the STOP state.
Resetting is also possible even when the RUN/STOP key switch of the CPU is set to the RUN position if the CPU is stopped by an error detectable by the self-diagnosis function.

POINT

Remote RESET cannot be executed when the CPU is in the RUN state.

(1) Application of remote RESET

Remote RESET can be used to reset the CPU by remote operation after an error has occurred in cases where the inaccessibility of the CPU module installation site makes direct operation impossible.

(2) Method for remote RESET

Remote RESET can only be performed by operations from GPPQ or a serial communication module.

(a) Regardless of whether reset is executed from GPPQ or a serial communication module, the setting to enable remote RESET must be made in the parameters before performing the reset operation. The remote RESET enable/disable setting is set in the PC system settings of the GPPQ parameter mode.

(b) After enabling remote RESET in the parameter settings, execute the reset by remote operation.

- When using GPPQ, use remote operation under the PC menu in any mode.
- When using a serial communication module, use dedicated protocol commands.

For details on serial communication module control, refer to the serial communication module User's Manual.

10.6.5 Remote latch clear

Remote latch clear is a function for resetting the latched device data of the QnACPU while the CPU is in the STOP state, e.g. by using GPPQ.

POINT

Remote latch clear cannot be executed when the CPU is in the RUN state.

(1) Application of remote latch clear

Remote latch clear is useful for performing latch clear in the cases indicated below. In this case, the function is used in combination with the remote RUN/STOP function.

- When the CPU is installed in an inaccessible location;
- Latch clear of a CPU in a control panel is to be executed from an external location.

(2) Method for remote latch clear

Remote latch clear can only be performed by operations from GPPQ or a serial communication module.

- For operation from GPPQ, remote operation is performed under the PC menu in any mode.
- RUN/STOP control using a serial communication module is achieved by using commands in the dedicated protocol.

For details on serial communication module control, refer to the serial communication module User's Manual.

POINTS

(1) According to the device latch ranges set in device setting in the parameter mode, there are ranges within which latch clear is valid and ranges within which it is not valid. Remote latch clear is only valid for devices set in the range for which "latch clear valid" is set.

(2) When remote latch clear is executed, devices that are not latched are also cleared.

10.6.6 Relationship between remote operation and CPU key switch

The operating status of the CPU is determined as follows in accordance with the combination of the remote operations described in sections 10.6.1 through 10.6.5 and the position of the RUN/STOP key switch of the CPU.

Remote Operation / Key Switch	RUN *1	STEP-RUN	STOP	PAUSE *2	RESET *3	Latch Clear
RUN	RUN	STEP-RUN	STOP	PAUSE	Operation in possible*4	Operation in possible*4
STOP	STOP	STOP	STOP	STOP	RESET	Latch clear

*1 If executed using a remote RUN contact, "RUN-PAUSE Contact" must be set in advance in PC system setting in the parameter mode.

*2 If executed using a remote PAUSE contact, "RUN-PAUSE Contact" must be set in advance in PC system setting in the parameter mode.
In addition, the remote PAUSE enabled coil must also be turned ON in advance.

*3 "Allow Remote Reset Yes" must be set in advance in PC system setting in the parameter mode.

*4 RESET is possible if the CPU is stopped by remote operation.

When the RUN/STOP key switch is set to RUN and multiple remote operation requests are received, the CPU performs the operation with the highest priority first.

Remote Operation	RUN	STEP-RUN	STOP	PAUSE	RESET	Latch Clear
Order of priority	(4)	(3)	(1)	(2)	—	—

The order of priority increases from (4) to (1).

10.7 Terminal Operation

This function sets a Q6PU programming unit in the terminal mode and performs the data communications shown below by using instructions for QnACPU peripheral devices.

(1) Display of messages from the QnACPU on the display of the Q6PU.

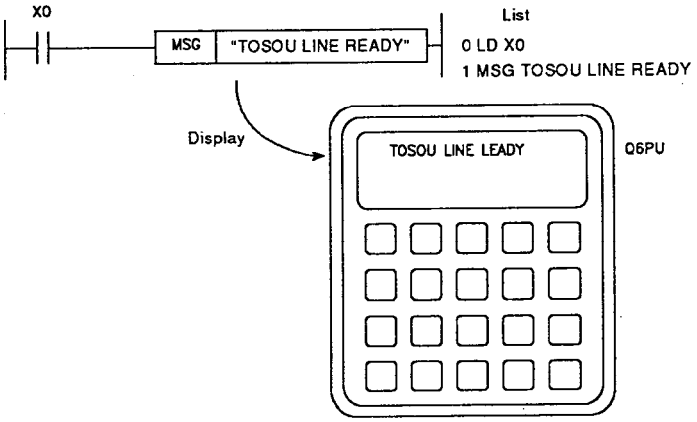
(2) Storage of Q6PU key input data in the devices of the QnACPU.

In this way, the Q6PU can be used as a terminal of the QnACPU.
This function is explained from the next section onward. However, for details on instructions for peripheral devices, refer to the QnACPU Programming Manual (Common Instructions).

10.7.1 Operation for message display

Designated character strings can be displayed at a Q6PU by using the peripheral device instruction MSG.
It is also possible to display character strings with GPPQ by using the CPU messages of the display menu in the PC mode.

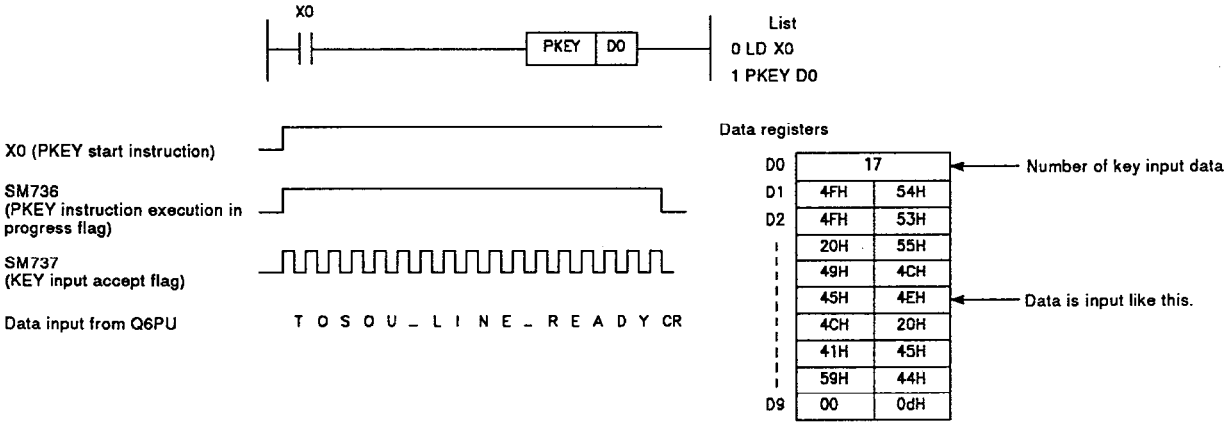
Example: • Program to display "TOSOU LINE READY" as message No.1 at the Q6PU when X0 is turned ON.



10.7.2 Key input operation

Character string data keyed in at a Q6PU can be stored without change as ASCII data in designated devices by using the peripheral device instruction PKEY.
Data input ends when a CR code is input or when the 32nd character is input.

Example: • Program to input "TOSOU LINE READY" at the Q6PU when X0 is turned ON.



10.8 Reading Module Access Time Intervals

The QnACPU can monitor the access interval time (the time between one access reception and the next access reception) for intelligent special function modules, network modules, data link modules, or GPPQ. This allows you to determine the frequency of accesses to the CPU from external sources.

The operation for reading the module access interval time involves the following special relay and special registers.

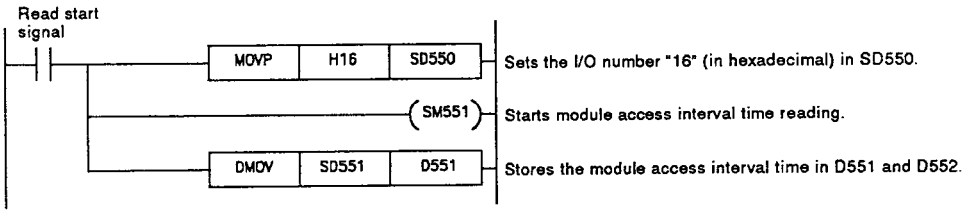
(1) Special relay

Number	Name	Description
SM551	Module access interval time read	When this relay turns from OFF to ON, the module access interval time for the special function module designated in special register SD550 is read into special registers SD551 to SD552. ON : Read OFF : No processing

(2) Special registers

Number	Name	Description
SD550	Module for module access interval time measurement	Set the I/O number of the module whose module access interval time is to be measured in this register. The I/O number of the peripheral device connected to the RS-422 interface of the CPU is HFFFF. The I/O number is set with the lowest digit omitted.
SD551 to SD552	Module access interval time	When SM551 is turned ON, it stores the interval time for access from the module designated at SD550. SD551: 1 ms units (range: 0 to 65535) SD552: 1 μs units (range: 0 to 900, stored every 100 μs) Example: When the module access interval time is 123.4 ms: SD551=123, SD552=400

Example program:
Program for reading the module access interval time of the special function module at X/Y160.



POINT

To read the access interval time for access from GPPQ at another station in the network, set the I/O number of the network module.

11. COMMENTS THAT CAN BE STORED IN QnACPU

11.1 Function List

The QnACPU can store various types of comment. This both makes the CPU easier to operate and makes it easier for persons other than program designers to read programs.
The types of comment that the QnACPU can stores are listed in the table below.

Item	Function	Refer to
PC name	Naming the CPU used.	Section 11.2
Drive title	Assigning a title to each drive.	Section 11.3
File title	Assigning a title to each file.	Section 11.4
Device comment	Assigning comments and labels to devices used in a program.	Section 11.5
Statements/notes	Assigning comments to each program step number or P, I pointer.	Section 11.6
Device initial value comment	Assigning a comment to the device initial value file.	Section 11.7

For details on the setting method for each function, refer to the SW□IVD-GPPQ Operating Manual (Online)/(Offline).

11.2 PC Name

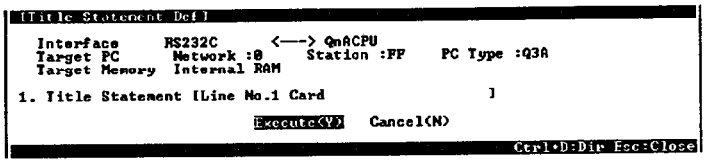
PC name appends a comment to the CPU to make it easier to confirm the accessed CPU when accessing the QnACPU from GPPQ.
Two types of PC name can be set: labels and comments. The settings are made on the "Define PC Name" screen in the parameter mode of GPPQ.

The settings to be made are indicated in the table below.

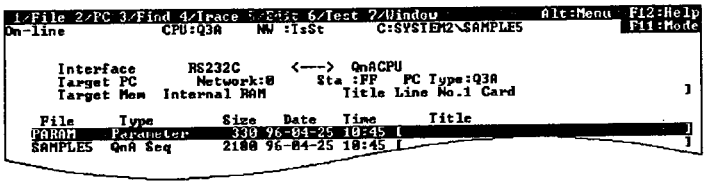
Item	Setting Details	Setting Range	Default Value
Label	Set the label for the CPU.	Max. 10 characters	No setting
Comment	Set the comment for the CPU.	Max. 64 characters	

11.3 Drive Title

The drive title function allows a title to be assigned to each drive to make it clear what files are stored in the internal memory and memory card.
Drive titles are created on the "Title Statement Def" screen under the PC menu in the online mode of GPPQ.



The created title is displayed on the screen as shown below.

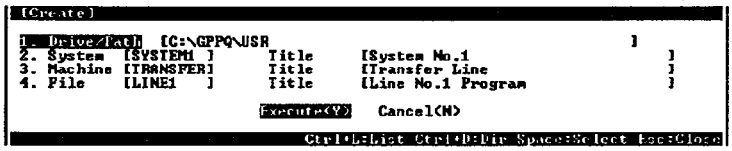


POINT

Note that, when drive titles are created, memory area equivalent to one file is used in each memory.

11.4 File Title

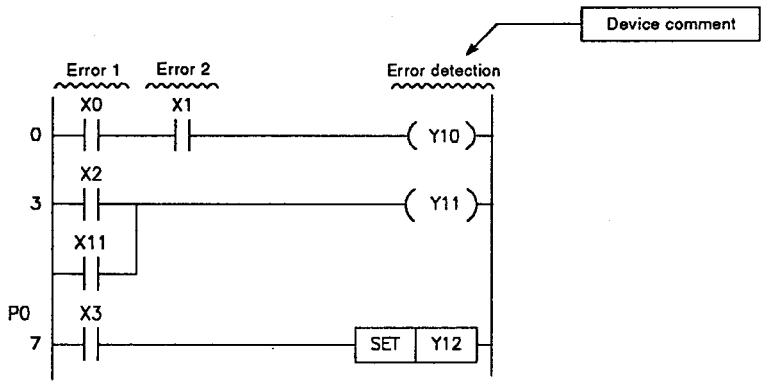
The file title function allows file titles to be assigned to files so that the contents of files can be determined.
File titles are set in file setting when starting up GPPQ, or in PC writing under the PC menu in any GPPQ mode. The maximum number of characters in a comment is 32.



File titles are stored in the created files themselves.

11.5 Device Comment

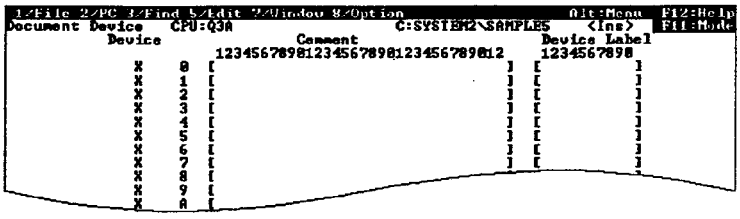
The device comment function allows comments to be assigned to devices to make programs easier to read.
In addition, if the CPU type is set to "Xtype" with GPPQ when creating a program, the program can be created using labels instead of devices.



POINT

A memory card is required to create device comments and store the device comment file in the CPU.

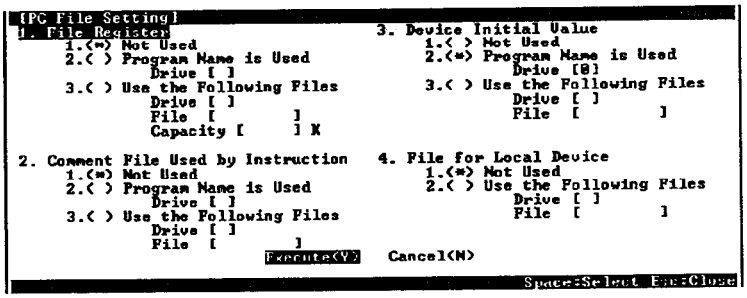
(1) Device comments are set in the documentation mode of GPPQ.
The maximum number of characters is 32 for comments and 10 for labels (device label names).



• The devices to which comments and labels can be assigned are listed below.

Device name: X, Y, M, L, F, SM, B, SB, V, T (present value), C (present value), ST (present value), D, SD, W, SW, R, ZR, P, I, U□\G□, J□\X, J□\Y, J□\B, J□\SB, J□\W, J□\SW, BL□\S, BL□\TR
(When P, I comments are used as pointers for subroutine programs, interrupt programs, etc., the comments are not displayed. To display one of these comments, express it as a pointer statement (see Section 11.6)).

(2) When comments are used with application instructions (LEDC, PRC, etc.) and device comment files have been written in the CPU, a parameter setting must be made to determine which is recognized as the device comment file.
This setting is made at "2. Comment File Used by Instruction" on the "PC File Setting" screen in the parameter mode of GPPQ.



The setting details are as follows.

1) "Not Used"
No setting is made for the comment file to be used. To use a comment file, use the QCDSET instruction. (For details on the QCDSET instruction, refer to the QnACPU Programming Manual (Common Instructions).)

2) Using the designated device comment file:
Use the file name stored in the drive designated by parameter.

3) "Program Name is Used"
Use the comment file with the same file name as the program that exists in the designated drive and is currently being executed. When the program is changed the comment file also changes.

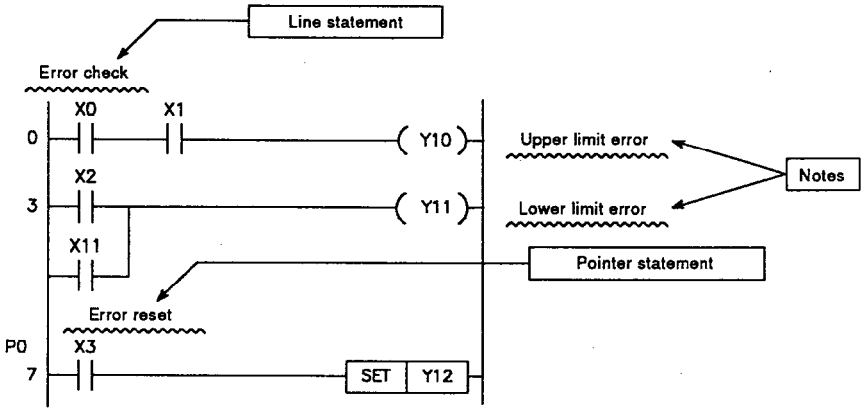
POINTS

(1) When using the QCDSET instruction, note the following points.
1. When 1) or 2) above has been set, the file set with the QCDSET instruction is valid with all program files.
2. When 3) above is set, the file set with the QCDSET instruction is valid only with the program file for which the QCDSET instruction is executed.

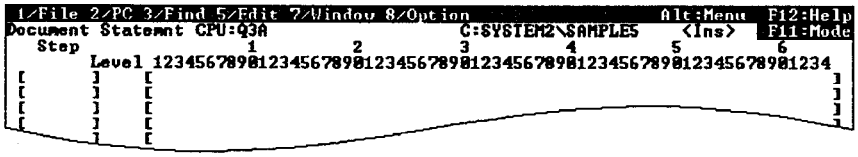
(2) Even if the file in the parameters does not exist at the designated drive, no CPU error occurs. Also, since no file exists, the CPU does not display comments.

11.6 Statements/Notes

Statements and notes are assigned to each program step, or to P, I pointers, in order to make the program easier to read.



(1) Statements and notes are set on the "Pointer statement", "Statement", and "Note" screens under the edit menu in the documentation mode of GPPQ.



(2) The applications of each of these types of comments are as follows.

(a) Statement (Line statement)

These comments can be appended to ladder blocks for particular functions in order to explain the significance and application of the function.

(b) Pointer statement

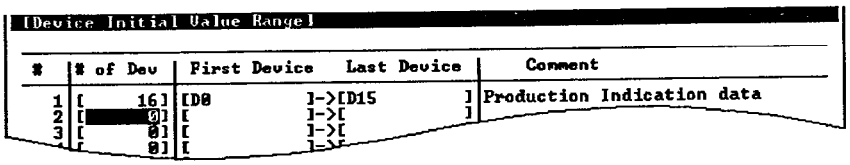
These comments can be appended to the pointers at the head of subroutine programs and interrupt programs to explain the significance and application of each program.

(c) Note

These comments can be appended to individual ladder blocks to explain the significance and application of their functions.

11.7 Device Initial Value Comment

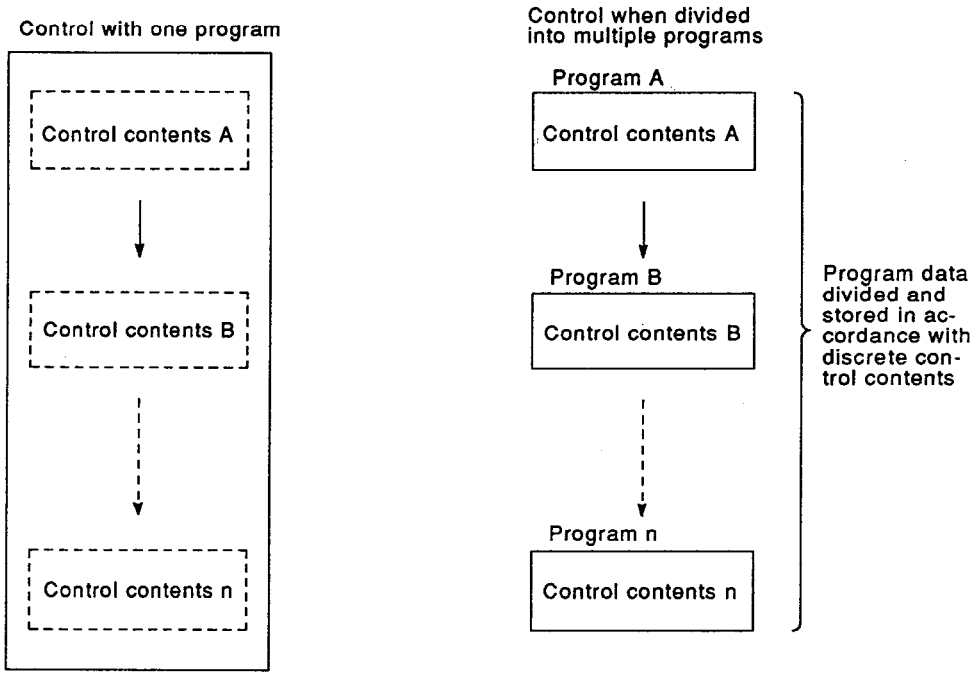
Device initial value comments are assigned to device initial value files to make it possible to determine the contents of the files.
Device initial value comments are stored in device initial value files.
They are set on the "Device Initial Value Range" screen under the edit menu in the device mode of GPPQ.



12. OVERVIEW OF PROCESSING PERFORMED BY THE QnACPU

12.1 Program Execution Types

The programs executed by the QnACPU are stored in the internal memory of the CPU or in a memory card.
It is possible to store program data grouped together as one program in the internal memory or memory card, but also possible to divide it into multiple programs based on control units.
For this reason when programming is undertaken by more than one designer, each designer can divide his programming into single processing units for storage in the CPU internal memory or memory card.

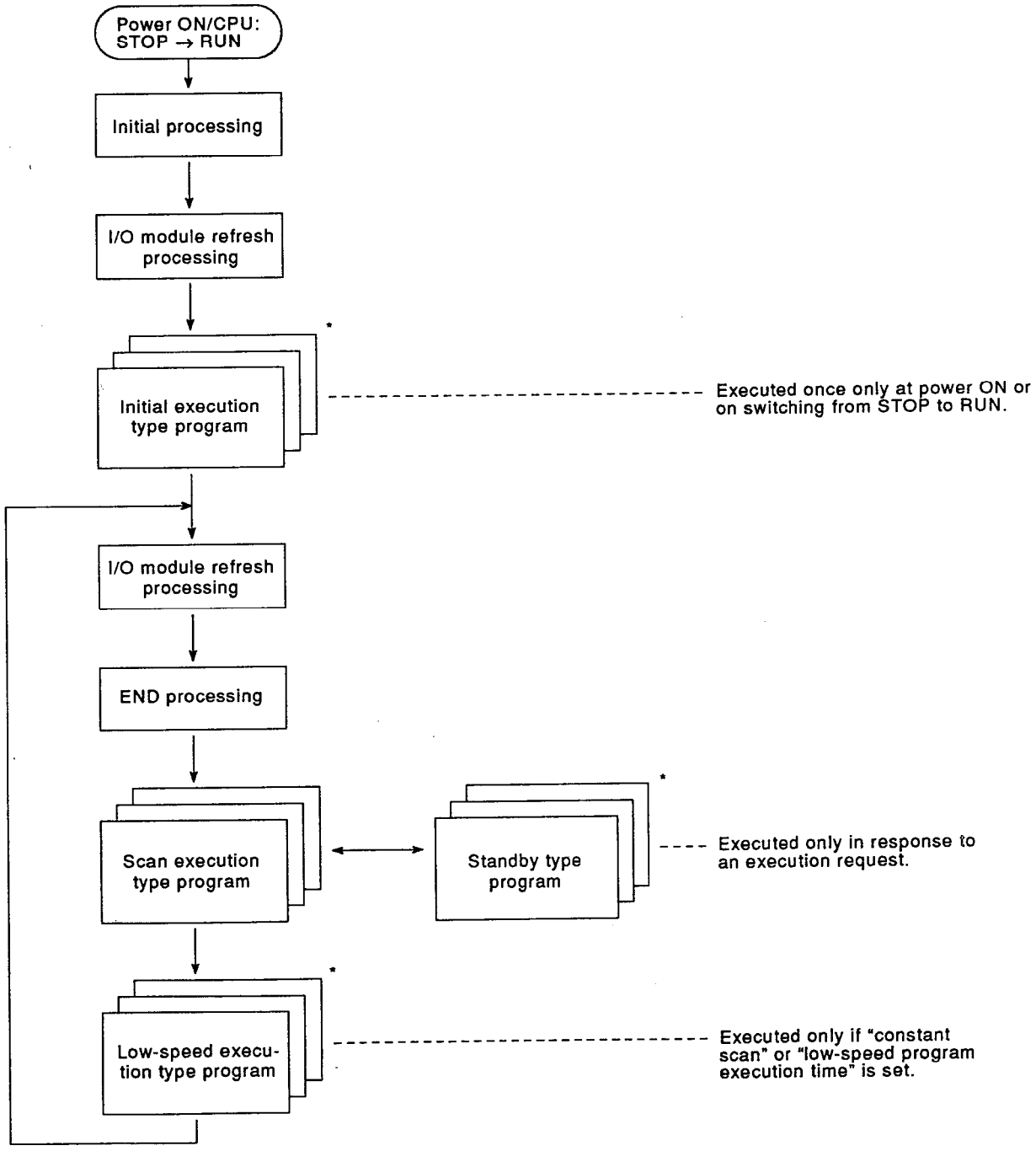


When the program data is to be divided into multiple programs, set their "execution type" in program setting in the parameter mode of GPPQ.
The QnACPU executes each program of the set execution types in the setting order.
There are four executions types "initial execution type", "scan execution type", "low-speed execution type", and "standby type".

- Initial execution type : Program that is executed once only when the power is turned ON or on switching from STOP to RUN.
  (See Section 12.1.1)
- Scan execution type : Program that is executed once per scan, starting from the scan following the one in which the initial execution type program is executed.
  (See Section 12.1.2)

- Low-speed execution type: Program that is executed only in the surplus scan time after execution of scan execution type programs when "constant scan" is set, or when a low-speed type program execution time is set. (See Section 12.1.3)
- Standby type : Program that is only executed when an execution request is made for it. (See Section 12.1.4)

The flow of operation processing when the QnACPU power is turned on or the CPU is switched from STOP to RUN is shown below.



POINT

With QnACPU, it is not necessary to set all execution types.
Use initial execution type programs, low-speed execution type programs, and standby type programs (all marked with an asterisk above) if required.

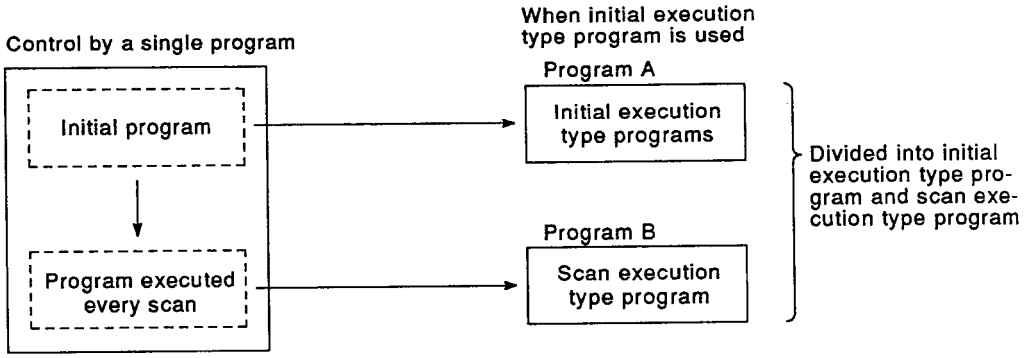
12.1.1 Initial execution type programs

(1) Definition

(a) An initial execution type program is a program that is executed once only when the power is turned ON or on switching the CPU from STOP to RUN.

(b) The execution type in program setting in the parameter mode of GPPQ is set as "Init".

(c) Initial execution type programs can be used for applications such as the initial processing for a special function module, where once the program has been executed once, there is no need to execute from the next scan onward.*

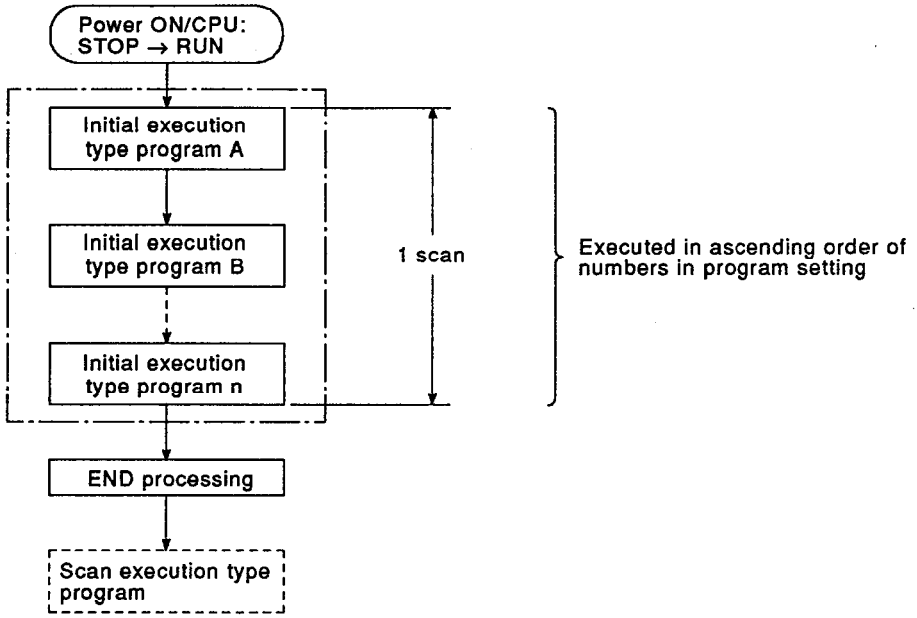


(2) Execution of multiple initial execution type programs

If there is more than one initial execution type program, they are executed in ascending order of the numbers set in program setting in the parameter mode.

(3) END processing

When execution of all initial execution type programs is completed, END processing is executed and a scan type execution program is executed from the next scan.



POINT

*: Instructions that contain a completion device cannot be used in initial execution type programs.

(4) Initial scan time

(a) This is the execution time of an initial execution type program. If multiple initial execution type programs are executed, it is the time taken to complete execution of all the initial execution type programs.

(b) The QnACPU measures the initial scan time and stores it in special registers SD522 and SD523.*1
The initial scan time can be checked by monitoring SD522 and SD523.

Example:
If "3" is stored in SD522 and "400" is stored in SD523, the initial scan time is 3.4 ms.

(5) Initial execution WDT time

This is a timer for monitoring the execution time of initial execution type programs; no default value is set.
To monitor the execution time of an initial execution type program, a setting within the range of 10 ms to 2000 ms can be made in "PC RAS setting" in the parameter mode (setting units: 10 ms).
If the initial scan time exceeds the set initial execution monitoring time, a "WDT ERROR" occurs and QnACPU stops operation.

POINTS

(1) *1: The accuracy of each scan time stored in the special registers is ±0.1 ms.
Note that even if a watchdog timer reset instruction (WDT) is executed in the sequence program, measurement of scan time is continued.

(2) An error in the range of 0 to 10 ms is generated in measurement of the initial execution monitoring time.
Because of this, if the initial execution monitoring time (t) is set as 10 ms, a WDT ERROR will occur with an initial scan time in the range of 10 ms ≤ t < 20 ms.

12.1.2 Scan execution type program

(1) Definition

(a) A scan execution type program is a program that is executed once every scan, starting from the scan following the one in which the initial execution type program was executed.

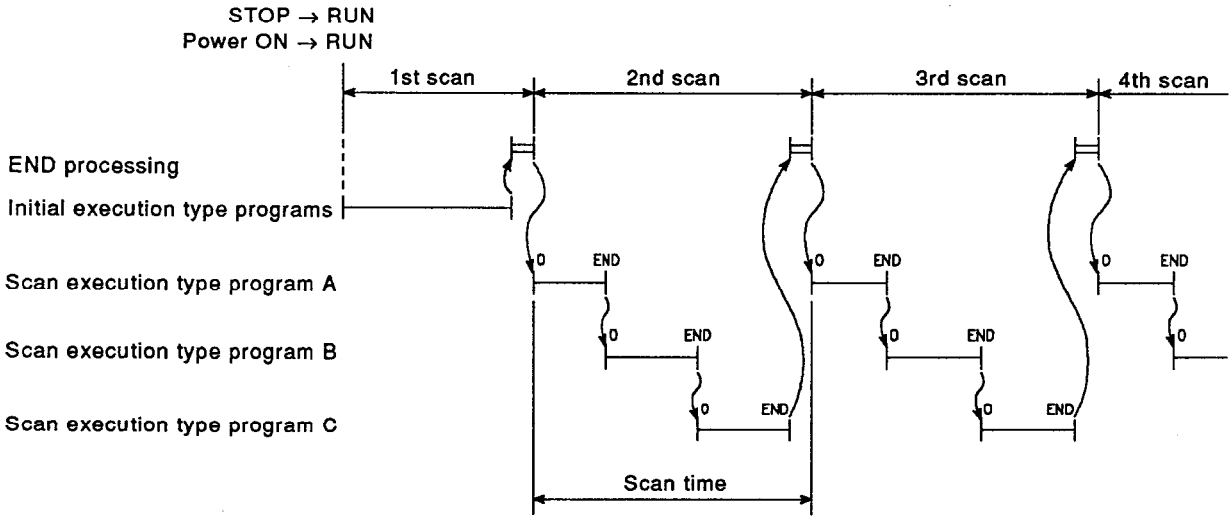
(b) The execution type in program setting in the parameter mode of GPPQ is set as "Scan".

(2) Execution of multiple scan execution type programs

If there is more than one scan execution type program, the programs are executed in ascending order of the numbers set for them in program setting in the parameter mode.

(3) END processing

When all the scan execution type programs have been executed, END processing is performed then the first scan execution type program is executed again.
By inserting a COM instruction at the end of a scan execution type program, END processing (general data processing, link refresh) can be executed for each program.



(4) When constant scan time is set *1

When constant scan is set, the scan execution type program is executed once every set constant scan time.

REMARK

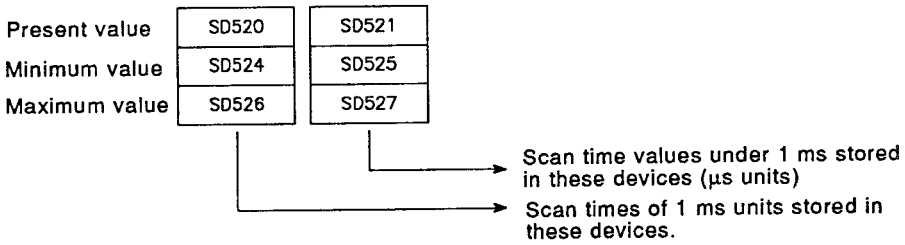
*1: Constant scan is a function whereby scan execution type programs are repeatedly executed in a fixed time interval.
(See Section 10.2).

POINT

For details on the index register processing performed when an interrupt program is executed during execution of a scan execution type program, refer to the QnACPU Programming Manual (Fundamentals).

(5) Scan time

- The scan time is the total time obtained by adding the execution time of the scan execution type program, the END processing time, and either the execution time for a low-speed execution type program or the constant scan waiting time.*1
  When more than one scan execution type program is executed, the "execution time of the scan execution type program" is the time taken to complete execution of all the scan execution type programs.
  *1: See Section 12.1.3.
- The QnACPU measures the present value, minimum value, and maximum value for scan time and stores them in special registers SD520, SD521, and SD524 to SD527.*2
  The scan time can be checked by monitoring these special registers.



Example:
If "3" is stored in SD520 and "400" is stored in SD521, the scan time is 3.4 ms.

(6) WDT (watchdog timer)

This is a timer that monitors scan time; its default value is 200 ms.
WDT is set within the range of 10 ms to 2000 ms in "PC RAS setting" in the parameter mode. (Setting units: 10 ms).
When low-speed execution type programs are used, make sure that WDT is set so that the value is greater than the sum of the scan time and the time needed for executing the low-speed execution type programs to be used.
If the scan time (total of execution times for scan execution type programs and low-speed execution type programs, and END processing time) exceeds the time set for WDT, a "WDT ERROR" occurs and the QnACPU stops operation.

POINTS

(1) *2: The accuracy of each scan time stored in the special registers is ±0.1 ms.
Even if the watchdog reset instruction (WDT) is executed in the sequence program, measurement of each scan time continues.

(2) The WDT measurement error is 0 to 10 ms.
Because of this, if WDT (t) is set to 10 ms, a WDT ERROR may not occur within the range 10 ms ≤ t < 20 ms.

12.1.3 Low-speed execution type program

(1) Definition

(a) A low-speed type execution program is a program that is only executed during the surplus scan time in constant scan operation or during the execution time set for low-speed execution programs.

- If it is required to keep scan time constant and give priority to accuracy in control, a constant scan time should be set in "PC RAS setting" in the GPPQ parameter mode.
  (Setting range: 5 to 2000 ms; units: 5 ms)
- To secure an execution time in each scan for low-speed execution type programs to ensure that low-speed execution type programs will definitely be executed, set a low-speed execution type program execution time in "PC RAS setting" in the parameter mode. (Setting range: 1 to 2000 ms; setting units: 1 ms)
- In order to execute low-speed execution type programs, either a constant scan time or a low-speed execution program execution time must be set.

(b) Set "Slow" as the execution type in program setting in the parameter mode.

(c) This execution type is used for programs that do not have to be executed every scan, such as printer output.

(2) Execution of multiple low-speed execution type programs

If there is more than one low-speed execution type program, they are executed in ascending order of the numbers set in program setting in the parameter mode.

(3) Execution time for low-speed execution type program executed in one scan

(a) If a constant scan time is set, a low-speed execution type program is executed during the surplus time in the constant scan time.
Accordingly, the execution time for low-speed execution type program is different from scan to scan.
However, if the surplus time in the constant scan time is less than 2 ms, the low-speed execution type program cannot be executed.
When a low-speed execution type program is used, set the constant scan time so as to ensure a surplus time of at least 2 ms.

(b) When a low-speed program execution time is set, the low-speed execution type program is executed during this low-speed program execution time.
Accordingly, the scan time is different from scan to scan.

(c) If it has not been possible to process the low-speed execution type program within the surplus time of the constant scan time or within the low-speed program execution time, program execution is temporarily stopped and the rest of the program is executed in the next scan.

POINTS

(1) For details on the index register processing for switching from a scan execution type program to a low-speed execution type program, refer to the QnACPU Programming Manual (Fundamentals).

(2) For details on the index register processing for executing an interrupt program during execution of a low-speed execution type program, refer to the QnACPU Programming Manual (Fundamentals).

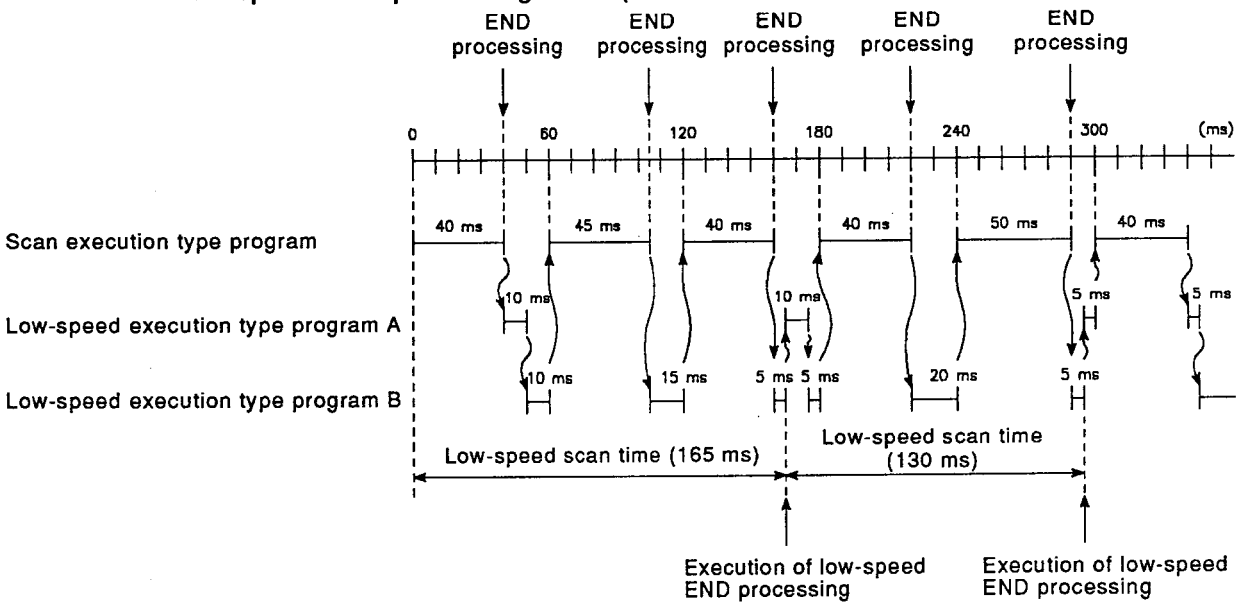
(3) When setting the low-speed program execution time, make sure that the sum of the scan time and the low-speed execution scan time is smaller than the set value for the WDT.

Example:

(1) When "constant scan" is set

The operation when a low-speed execution program is executed under the following conditions is shown below.

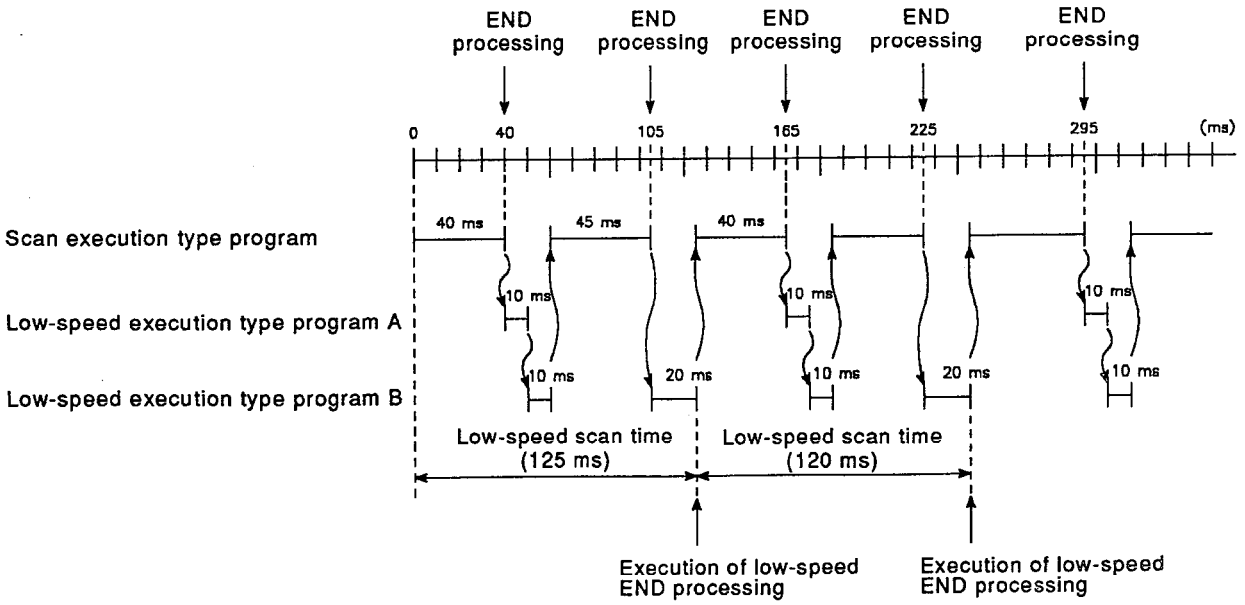
- Constant scan time: 50 ms
- Total for scan execution type programs: 40 ms to 50 ms
- Execution time for low-speed execution type program A: 10 ms
- Execution time for low-speed execution type program B: 30 ms
- END processing: 0 ms (made 0 ms here to make the explanation easy)
- Low-speed END processing: 0 ms (made 0 ms here to make the explanation easy)



(2) When a low-speed program execution time is set

The operation when a low-speed execution program is executed under the following conditions is shown below.

- Low-speed program execution time: 20 ms
- Total for scan execution type programs: 40 ms to 50 ms
- Execution time for low-speed execution type program A: 10 ms
- Execution time for low-speed execution type program B: 30 ms
- END processing: 0 ms (made 0 ms here to make the explanation easy)
- Low-speed END processing: 0 ms (made 0 ms here to make the explanation easy)



(4) END processing

When all low-speed execution type programs have been executed, low-speed END processing is executed.
The following processing is executed in low-speed END processing:

- Setting of special relays/special registers for low-speed programs
- Write during RUN of low-speed execution programs
- Measurement of low-speed scan time
- Resetting of watchdog timer for low-speed execution type programs

When low-speed END processing is completed, the first low-speed execution type program is executed again.

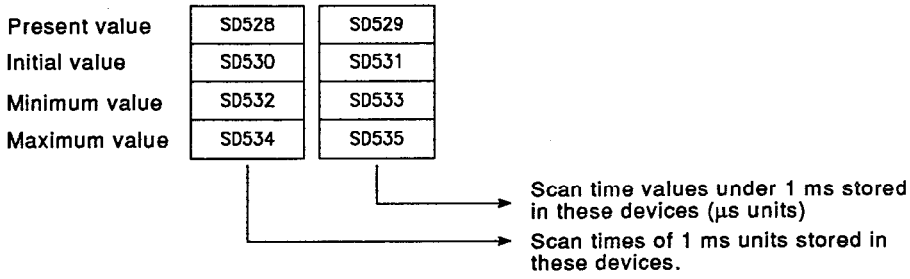
POINT

When a low-speed execution type program is executed, the constant scan time may be extended by a time equivalent to the maximum processing time for the instructions executed plus the low-speed END processing time.

(5) Low-speed scan time

(a) The low-speed scan time is the total time obtained by adding the time taken to execute the low-speed execution program and the low speed END processing time.
If multiple low-speed execution type programs are executed, it is the total time obtained by adding the time taken to execute all the low-speed execution programs and the low speed END processing time.

(b) The QnACPU measures the low-speed scan time and stores it in special registers SD528 to SD535.*1
The low-speed execution scan time can be checked by monitoring these registers.



Example:
If "3" is stored in SD528 and "400" is stored in SD529, the scan time is 3.4 ms.

(6) Low-speed execution WDT time

This is a timer that monitors the execution time of low-speed execution type programs; no default value is set.
To monitor the execution time of low-speed execution type programs, set this timer within the range of 10 ms to 2000 ms in "PC RAS setting" in the parameter mode. (Setting units: 10 ms).
If the low-speed scan time exceeds the set low-speed execution WDT time, a "PRG TIME OVER" error occurs. However, QnACPU operation continues.

POINTS

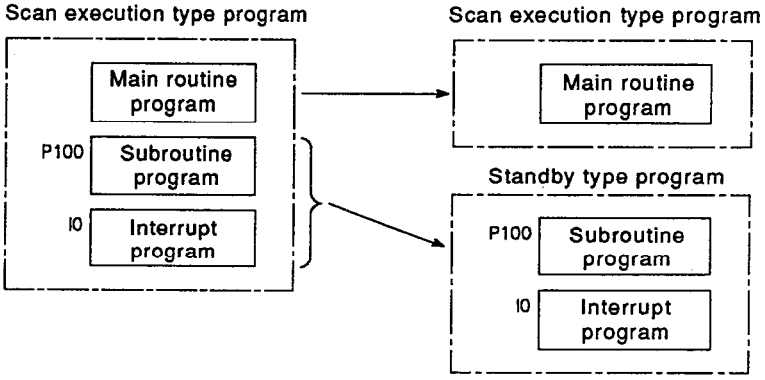
(1) *1: The accuracy of each scan time stored in the special registers is ±0.1 ms.
Even if the watchdog timer reset instruction (WDT) is executed in the sequence program, measurement of each scan time continues.

(2) The low-speed execution WDT time is measured in low-speed END processing.
Because of this, if the low-speed execution WDT time (t) is set at 100 ms, a PRG TIME OVER error occurs if the low speed scan time measured in low-speed END processing exceeds 100 ms.

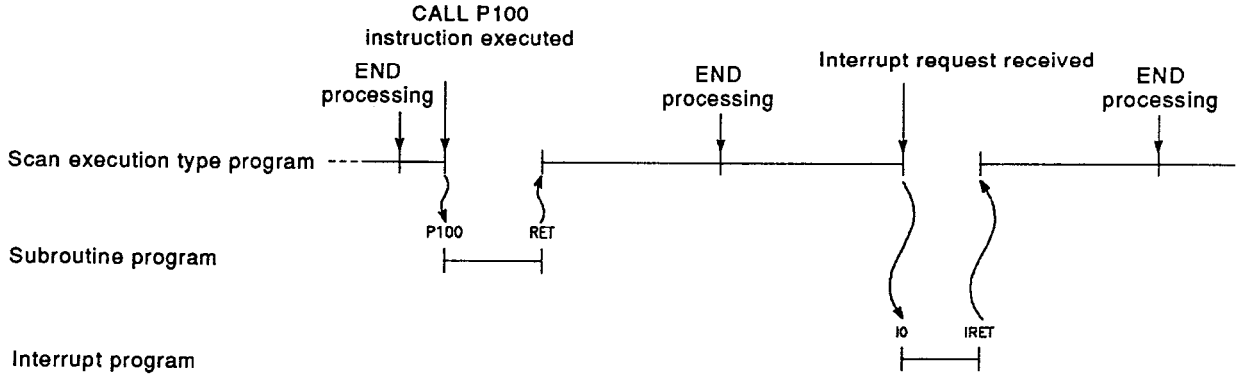
12.1.4 Standby type program

(1) Definition

(a) Standby type programs are programs that are only executed in response to an execution request.
They are used to control subroutine programs and interrupt programs separately from the main routine program.
It is possible to create multiple subroutine programs and interrupt programs for one standby type program.



(b) On completion of execution of a standby type execution program, execution returns to the program that was being executed before execution of the standby program.
The operation when a subroutine program and interrupt program of a standby type program are executed is shown below.



POINTS

(1) Since present values of timers are updated, and their contacts turned ON/OFF when the OUT T□ instruction is executed, they cannot be used in standby type programs.

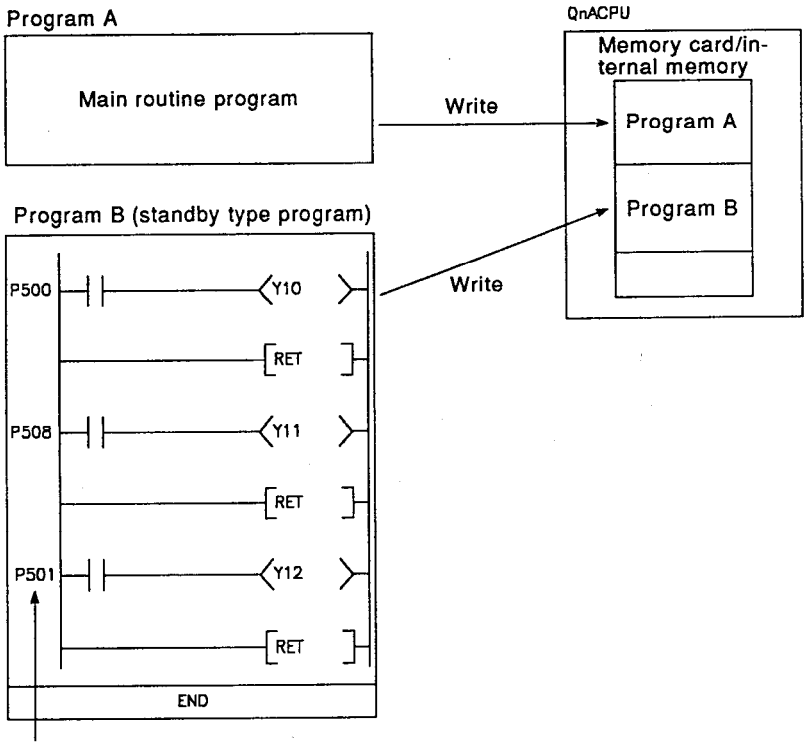
(2) When setting a subroutine program as a standby type program, use a common pointer.
Standby type programs for which local pointers are used cannot be executed.
For details on common pointers and local pointers, refer to the QnACPU Programming Manual (Fundamentals).

(2) When subroutine programs are integrated into one program

(a) Create subroutine programs in order starting from step 0 of the standby program.
An END instruction is required at the end of the subroutine program.

(b) Since there are no restrictions on the order of creation of subroutine programs, there is no need to arrange pointers in ascending order of pointer numbers when creating multiple subroutine programs.

(c) Use a common pointer.*
Subroutine programs with which common pointers are used can be called from all programs that can be executed by the QnACPU.



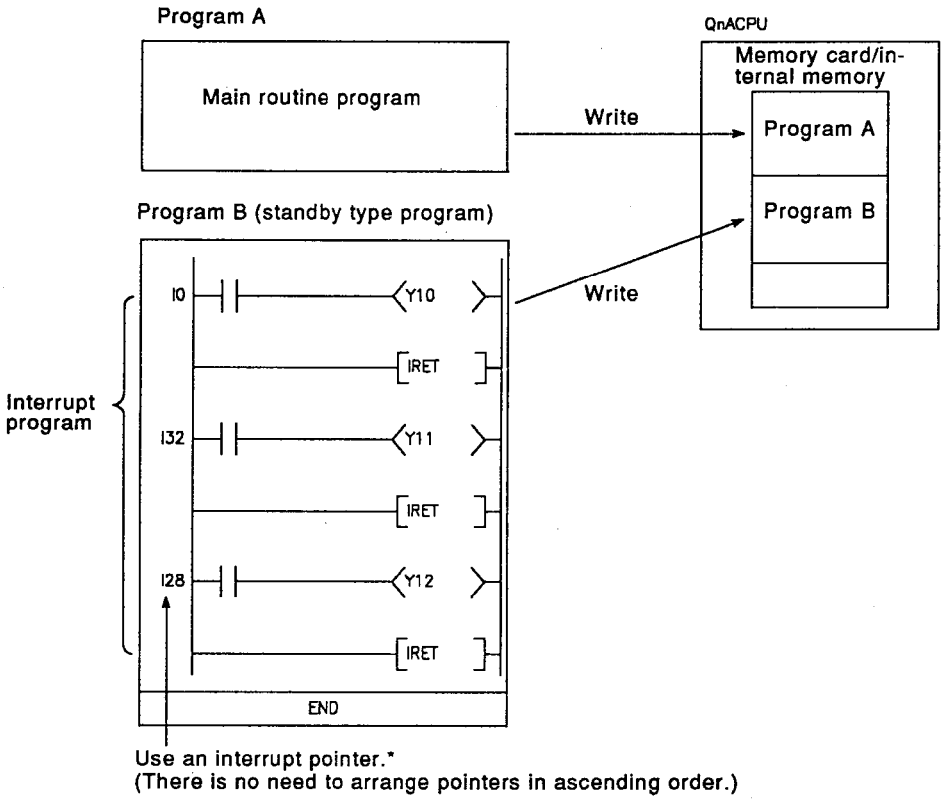
REMARK

*: For details on common pointers, refer to the QnACPU Programming Manual (Fundamentals).

(3) When interrupt programs are integrated into one program

(a) Create interrupt programs in order starting from step 0 of the standby program.
An END instruction is required at the end of the interrupt program.

(b) Since there are no restrictions on the order of creation of interrupt programs, there is no need to arrange pointers in ascending order of pointer numbers when creating multiple interrupt programs.



REMARK

*: For details on interrupt pointers, refer to the QnACPU Programming Manual (Fundamentals).

12.1.5 Initial processing

This is the pre-processing required to execute sequence operation, and is executed once only when the power is turned ON, when a reset operation is performed, and on switching the CPU status from STOP to RUN.

At power ON/reset

(a) The I/O modules are reset and initialized.

(b) The range in the data memory not subject to latch setting is initialized (i.e., bit devices are turned OFF, word devices are set to "0").

(c) Of the items in the self-diagnosis performed by the PC CPU, the checks carried out on turning the power ON or resetting are performed. (See Section 9.3).

(d) I/O addresses are automatically allocated to I/O modules in accordance with the positions at which they are mounted on the main base unit and extension base unit.

(e) In the case of a MELSECNET/10 control station or MELSECNET(II)/B master station, the network/link parameter information is set for the network/data link modules and network communication/data link communication is started.

(f) If no boot designated has been made, the designated file is transferred from the memory card to the internal memory.

STOP → RUN

(a) Same as (d) above.

(b) Same as (e) above.

(c) Same as (f) above.

(d) If a device initial value file is set, the device initial values set in this file are set for the object devices.

12.1.6 I/O module refresh processing

I/O module refresh processing is performed as described in the QnACPU Programming Manual (Fundamentals).

12.1.7 END processing

This is the post-processing to terminate one operation processing of a sequence program and return execution to step 0 of the program.

(a) Blown fuse, I/O module verification, and low battery voltage self-diagnosis checks are performed (see Section 9.3).

(b) If there is a data read or write request from a peripheral device, computer link module (AJ71UC24, AJ71C24(S3), AD51(S3) etc.), or serial communication module (AJ71QC24), data communication is carried out between the PC CPU and the peripheral device, computer link module, or serial communication module.

(c) If there is a refresh request from a network/refresh module, refresh processing is executed.

(d) If the trace point for sampling trace is set in every scan (after END processing), the set device statuses are stored in the sampling trace area.

(e) The MELSECNET/MINI-S3 automatic refresh setting refresh processing is performed (see Chapter 7).

POINTS

- If the constant scan function (see Section 10.2) is set, the END processing time result is retained during the period between completion of END processing and commencement of the next scan.
- If a low-speed execution type program (see Section 12.1.3) is executed, low-speed END processing is executed separately from normal END processing.
  Low-speed END processing sets the special relays and special registers used for low-speed execution programs.

12.2 Operation Processing for RUN, STOP, PAUSE, STEP RUN

The QnACPU has four operating states: RUN, STOP, PAUSE, and STEP RUN.
This section describes the PC CPU operation processing in each of these operating states.

(1) Operation processing for RUN state

(a) The RUN state is the status in which sequence program operation is executed repeatedly according to the following sequence: step 0 → END (FEND) instruction → step 0.

(b) On entering the RUN state, depending on the setting in the parameters for the output mode on switching from STOP to RUN, the output statuses stored when the last STOP was executed are output.

(c) The processing time from the switch from STOP to RUN to sequence program operation start differs according to the system configuration but is normally 1 to 3 seconds.
However, it may be longer than this depending on the conditions.

(2) Operation processing for STOP state

(a) The STOP state is the status in which sequence program operation is suspended due to RUN/STOP key switch operation or a remote STOP (see Section 10.6.1).

(b) On entering the STOP state, the output statuses are stored and all outputs are turned OFF. With the exception of output (Y) data, data in data memories is retained.

(3) Operation processing for PAUSE state

(a) The PAUSE state is the status in which sequence program operation is suspended while retaining output and data memory statuses (see Section 10.6.3).

(4) Operation processing for STEP RUN

(a) STEP RUN is the operation mode in which sequence program operation processing can be suspended/continued in increments of single instructions, using GPPQ (see Section 8.7).

(b) Since operation processing is suspended while retaining output and data memory statuses, execution statuses can be confirmed.

(5) QnACPU operation processing at RUN/STOP switching

QnACPU Operation Processing / RUN/STOP Key Switch Operation	Sequence Program Operation Processing	External Outputs	Data Memory (Y, M, L, S, T, C, D)	Remark
RUN → STOP	Operation executed to END instruction then stopped.	Output statuses stored by OS; all outputs turned OFF.	Maintains the status immediately before STOP state established.	
STOP → RUN	Operation started.	Determined by parameter setting for output mode on switching from STOP to RUN.	Operation executed from the status immediately before STOP state established.	

POINT

The QnACPU executes the following processing regardless of whether the RUN, STOP, or PAUSE state is in effect.

- I/O module refresh processing
- Data communication with peripheral devices, computer link modules, and serial communication modules.
- Link refresh processing

Consequently, even if the CPU is set to the STOP or PAUSE state, I/O monitor and test operations using a peripheral device, reading/writing from computer link modules and serial communication modules, and communication with other stations using MELSECNET, are still possible.

12.3 Operation Processing When a Momentary Power Interruption Occurs

The QnACPU detects a momentary power interruption when the input power supply voltage supplied from the power supply module falls below the stipulated range.
When the QnACPU detects a power interruption, the following operation processing is performed.

(1) When a momentary power interruption shorter than the allowable momentary power interruption time occurs

(a) When a momentary power interruption occurs, output statuses are held and operation processing is suspended after the file name and error history has been stored.
(The timer count continues.)

(b) If there is an SFC continuous operation designation, system save processing is executed.

(c) If an interrupt program to be run in the event of AC DOWN (interrupt pointer I33) has been written, this interrupt program is run.
Create interrupt programs so that the scan time does not exceed 10 ms.
(If an A63P is used as the power supply module, the allowable momentary interruption time is 1 ms, which means that interrupt programs cannot be used.)

(d) When power is recovered after a momentary power interruption, operation processing continues.

(e) Even if a momentary power interruption occurs and operation is suspended, watchdog timer (WDT) timing continues. For example, if the WDT setting in the GPPQ parameter mode is 200 ms, and assuming that the scan time is 190 ms, a momentary power interruption of 15 ms will cause a watchdog timer error.

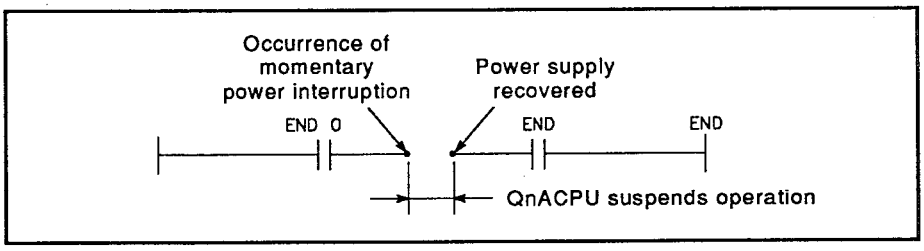


Fig. 12.1 Operation Processing on Occurrence of a Momentary Power Interruption

(2) When a momentary power interruption longer than the allowable momentary power interruption time occurs

The QnACPU performs an initial start.
The same operation processing occurs on resetting by turning the power ON or using the reset switch.

12.4 Data Clear Processing

When the power supply of the QnACPU is reset by turning the RUN/STOP key switch from RUN/STOP to RESET, or by turning the power OFF then back ON, clear processing is performed for all data except the following:

(1) Data in the internal memory (except data designated for memory clear in the boot designations) *1

(2) Data in the memory card

(3) Data for which a latch designation has been made

(4) Fault history data (with special relay SD storage)

*1: For details on the boot designations, see the QnACPU Programming Manual (Fundamentals).

Data (3) and (4) can be cleared by a latch clear operation with the RUN/STOP key switch (turn the switch four times to the L.CLR position) or by a remote latch clear operation from GPPQ (see Section 10.6.5).

Device latch designations are made by setting a latch range for each device on the "Device Setting" screen. There are two types of latch range setting.

(1) Latch clear key valid : Used to set a latch range which can be cleared by a latch clear operation using the key switch.

(2) Latch clear key invalid: Used to set a latch range which cannot be cleared even by a latch clear operation using the key switch.

Devices for which a latch clear key invalid setting is made can be cleared by using an instruction or a clear operation from the GPPQ.

Clear operation using an instruction	: Reset by RST instruction, or transfer K0 with MOV instruction.
Clear operation using GPPQ	: Execute a device memory all clear under the PC menu in the online mode.

For details on device latch ranges, refer to the QnACPU Programming Manual (Fundamentals).
For details on the operations when using GPPQ, refer to the SW□IVD-GPPQ Operating Manual (Online)/(Offline).

13. PARAMETER LIST

The parameters set when using QnACPU are listed in the table below. For details on each of the parameters, refer to the section or reference manual indicated.

Item		Description
PC name definition		Sets the label and comment for use by a peripheral devices for the CPU. This setting has no influence on CPU operation.
Label		Sets the CPU label.
Comment		Sets the CPU comment.
PC system setting		Sets the various settings required for the CPU system.
Timer interval	Low-speed timer	Sets the time limits for low-speed and high-speed timers.
	High-speed timer	
RUN-PAUSE contact		Sets the contact to control CPU RUN/PAUSE.
Remote reset		Sets enabled/disabled for remote reset operation.
Output at STOP → RUN		Sets the output mode on switching from STOP to RUN.
Common pointer No. from		Sets the first common pointer number.
General data process		Sets the number of modules processed in one data processing.
Number of free slots		Sets the number of points occupied by a vacant slot.
System interrupt	Interrupt counter	Sets the first interrupt counter number, and the fixed interval for the interrupt pointer.
	Fixed interval	
PC file setting		Sets the file types used by the CPU.
File register		Sets the file register file used.
Comment file used by instruction		Sets the comment file used with instructions.
Device initial value		Sets the file for initial device values to be used.
File for local device		Sets the file for local devices to be used.
Device setting		Sets the number of points, latch range, etc., for each device.
Devices		Sets the number of device points used.
Latch range (Enable C/L key)		Sets the latch range for which latch clear key operation is valid.
Latch range (Disable C/L key)		Sets the latch range for which latch clear key operation is invalid.
Local device		Sets the range of devices to be set as local devices.

Setting Details		Reference Section/Reference Manual
Default Value	Setting Range	
—	—	Section 11.2
No setting	Max. 10 characters	
No setting	Max. 64 characters	
—	—	—
100 ms	10 ms to 1000 ms (in 10 ms units)	QnACPU Programming Manual (Fundamentals)
10 ms	1 ms to 100 ms (in 1 ms units)	
No setting	X0 to X1FFF	Sections 10.6.1 and 10.6.3
Disabled	Enabled/disabled	Section 10.6.4
Before operation/After 1 scan	Before operation	Section 10.4
No setting	P0 to P4095	QnACPU Programming Manual (Fundamentals)
1 module	1 module to 6 modules	Section 6.3
16 points	0 to 64 points (16 point units)	Section 5.3
No setting	C0 to C65535	QnACPU Programming Manual (Fundamentals)
128 → 100 ms 129 → 40 ms 130 → 20 ms 131 → 10 ms	1 ms to 1000 ms (in 5 ms units)	
—	—	—
Not designated by parameter setting.	• Not designated by parameter setting. • Use same file name as program. • Not created.	QnACPU Programming Manual (Fundamentals)
Not designated by parameter setting.	• Not designated by parameter setting. • Use same file name as program. • Use the designated file.	Section 11.5
Use same file name as program.	• Not designated by parameter setting. • Use same file name as program. • Use the designated file.	QnACPU Programming Manual (Fundamentals)
Not designated by parameter setting.	• Not designated by parameter setting. • Use the designated file.	QnACPU Programming Manual (Fundamentals)
—	—	—
X → 8 k points Y → 8 k points M → 8 k points L → 8 k points B → 8 k points F → 2 k points SB → 2 k points V → 2 k points S → 8 k points T → 2 k points ST → 0 k point C → 1 k point D → 12 k points W → 8 k points SW → 2 k points	Fixed for X (8 k points), Y (8 k points), S (8 k points). Settings can be made within a range of 28.8 k words, including the numbers of points indicated above, with a maximum of 32 k points for each device. However, the total for bit devices is 64 k points.	QnACPU Programming Manual (Fundamentals)
No setting	1 range only for each device	Section 10.3
No setting	1 range only for each device	Section 10.3
No setting	1 range only for each device	QnACPU Programming Manual (Fundamentals)

Item		Description
PC RAS setting		Sets the settings for the RAS function.
Set WDT	WDT settings	Sets the watchdog timer for the CPU.
	Initial execution WDT time	
	Low-speed execution WDT time	
Error check		Sets detection or non-detection of designated errors.
Operation mode at error		Sets the CPU operation mode when an error is detected.
Constant scan		Sets the constant scan time.
Annunciator display mode	F No. display	Sets the display mode when an annunciator comes ON.
	Comment display	
	Time of occurrence	
Fault history		Sets the storage destination for the CPU fault history.
Slow program execution time		Sets the time for execution of low-speed execution type programs.
I/O allocation		Sets the mounting status of each module.
Slot settings	Type	Sets the type, number of points, head I/O No., etc., for modules.
	Number of points	
	Head XY	
	Model Name	
Base settings	Power supply module model name	Sets the power supply module model name and extension cable model name. This setting has no influence on CPU operation.
	Extension cable model name	
MELSECNET (II, /10) setting		Sets the link parameters for a MELSECNET II data link system or the network parameters for a MELSECNET/10 network system.
MELSECNET/MINI setting		Make the settings for automatic refresh of a MELSECNET/MINI system.
Number of master modules		Sets the number of MELSECNET/MINI master modules used.
MELSECNET/MINI detailed settings	Master module head I/O No.	Set the detailed setting required for automatic refresh of a MELSECNET/MINI system.
	Model name & number of stations	
	Receive data batch refresh	
	Send data batch refresh	
	Retry count for communication errors	
	FROM/TO instruction access priority	
	Receive data clear at communication error	
	Faulty station detection bit data	
	Error No.	
	MINI link operation when CPU stopped	
	Line error check	
Auxiliary settings		Sets the settings required when multiple programs are used.
Program		Sets the programs among multiple programs to be executed.
Boot		Set the file and other settings for boot operation.
SFC		Sets the settings required for SFC programs.
X/Y allocation confirm		Allows confirmation of the settings made in I/O allocation. This setting has no influence on CPU operation.

Setting Details		Reference Section/Reference Manual
Default Value	Setting Range	
—	—	—
200 ms	10 ms to 2000 ms (in 10 ms units)	Section 9.2
No setting	10 ms to 2000 ms (in 10 ms units)	Section 12.1.1
No setting	10 ms to 2000 ms (in 10 ms units)	Section 12.1.3
Check executed	Error check executed	Section 9.3
Stop	Stop/Operation continues	Section 9.3
No setting	5 ms to 2000 ms (in 5 ms units)	Section 10.2
Displayed	Displayed/not displayed	Section 9.8.2
Not displayed	Displayed/not displayed	
Not displayed	Displayed/not displayed	
Internal memory	Internal memory/designated history file	Section 9.4
No setting	5 ms to 2000 ms (in 5 ms units)	QnACPU Programming Manual (Fundamentals)
—	—	—
No setting	Vacant/input/output/special	Section 5.3
No setting	0 point to 64 points (16-point units)	
No setting	0 to 1FFF (in 10H units, hexadecimal)	
No setting	Max. 16 characters	
No setting	Max. 16 characters	Section 5.3
—	Refer to MELSECNET/10 Network System Reference Manual for QnA	MELSECNET/10 Network System Reference Manual for QnA MELSECNET, MELSECNET/B Data Link System Reference Manual
—	—	Section 7
0 module	0 to 8 modules	
No setting	Number of CPU I/O points	
MINIS3	MINIS3/MINI( ) stations	
X1000 to 200H	X, M, L, B, T, ST, C, D, W, R, ZR, none (bit devices designated in multiples of 16)	
Y1000 to 200H	Y, M, L, B, T, ST, C, D, W, R, ZR, none (bit devices designated in multiples of 16)	
5 retries	0 to 32 retries	
CPU priority	CPU priority/link priority	
Clear	Clear/hold	
No setting	M, L, B, T, ST, C, D, W, R, ZR, none	
No setting	D, W, T, ST, C, R, ZR	
Stop	Operation continues/stop	
Latch data	Test data/OFF data/latch data	
—	—	QnACPU Programming Manual (Fundamentals)
No setting	Program name, scan/low-speed/initial/standby	
No setting	File name, type, transfer source drive, transfer destination drive	
—	QnACPU Programming Manual (SFC)	QnACPU Programming Manual (SFC)
—	—	SW□IVD-GPPQ Operating Manual (Offline)

14. SELECTING MEMORY CARD MODELS

Since the QnACPU is provided with an internal memory to store parameters and programs as a standard feature, programs can be executed without installing a memory card.
The program capacities of the internal memories of the various models of CPU module are indicated below.

Q2ACPU ..... 28 k steps (112 k byte internal memory)

Q2ACPU-S1 ..... 60 k steps (240 k byte internal memory)

Q3ACPU ..... 92 k steps (368 k byte internal memory)

Q4ACPU ..... 124 k steps (496 k byte internal memory)

14.1 Applications of Memory Cards

A memory card is required in the following cases:

(1) To perform a boot operation

Parameters, programs, device initial values, separate comments, and boot files are stored in the memory card; on program execution they are read to internal memory and executed.

(2) To use file registers.(*1)

(3) To use local devices. (*2)

(4) To use the sampling trace function. (*2)

(5) To use the status latch function. (*2)

(6) To use the program trace function. (*2)

(7) To store the fault history in a file. (*2)

(8) To execute the maximum number of steps possible with a QnACPU.

When a program of the maximum capacity is stored in the internal memory, the parameter files and device initial values must be stored in a memory card.

(9) To use the SFC trace function. (*2)

14.2 Differences Between E^2PROM and Flash Memory

(1) E^2PROM

- Installed in the CPU; writing possible.
- Writing in file units is possible.

(2) Flash memory

- Writing not possible installed at the CPU.
- Installed at a peripheral device; only batch writing of files is possible (see Section 3.1).

*1: Can be set in the ROM area of the memory card if program file registers are read-only.

*2: Can only be set in the RAM area of the memory card.

14.3 Selecting Memory Card Capacity

Select the memory card capacity according to the type of files to be stored in the memory card and the size of these files. The sizes of files is calculated using the formulae presented below.

Function	Approximate File Capacity (Units: Bytes)
Drive title	64
Entry Password	72
Parameters *3	MELSECNET, NET/10 None → 330 When MELSECNET (II, /B) set → max. 4096 per module
Boot file	(Number of files x 18) + 67
Sequence program *3	(Number of steps x 4) + 122
Device comments *3	(34 x number of comment points) + (device types*1 x 10) + 64
Device initial values *3	(Number of device points x 2) + (device types x 44) + 66
File registers	Number of file register points x 2 bytes
Simulation data	(Number of word device points x 2) + + (number of bit device points ÷ 16) x 2 + (device ranges*2 x 44) + 66 Rounded up
Sampling trace data	(Added information + (number of word devices x 2) + + (number of bit devices ÷ 16) x 2) x trace count + (number of device points x 12) + 362 Rounded up
Status latch data	For all devices : 58576 For detailed devices : (Number of word device points x 2) + + (number of bit device points ÷ 16) x 2 + (device types x 8) + 352 Rounded up
Program trace data	Same as sampling trace
Fault history data	54 x number of faults stored + 72 bytes
SFC trace data	Max. 48 k (in 1 kbyte units)

*1: "Device types" means the number of registered device names.
For example, if D, W, and T are registered, it is 3 types.
*2: "Device ranges" means the number of registered range settings.
*3: These are the files transferred from the memory card to the internal memory in the boot operation.

POINT

Note that capacities will be secured by rounding up in the units indicated below according to the memory area used for storage:

Internal memory..... 4096 bytes (1 k step) units

Memory card..... 512 bytes units

Note also that when files are transferred from the memory card to the internal memory in boot operation, the capacity secured after transfer is different from that occupied before transfer.

15. HARDWARE SPECIFICATIONS OF THE CPU MODULE

15.1 General Specifications

The common specifications for each type of module are indicated in the table below.

General Specifications

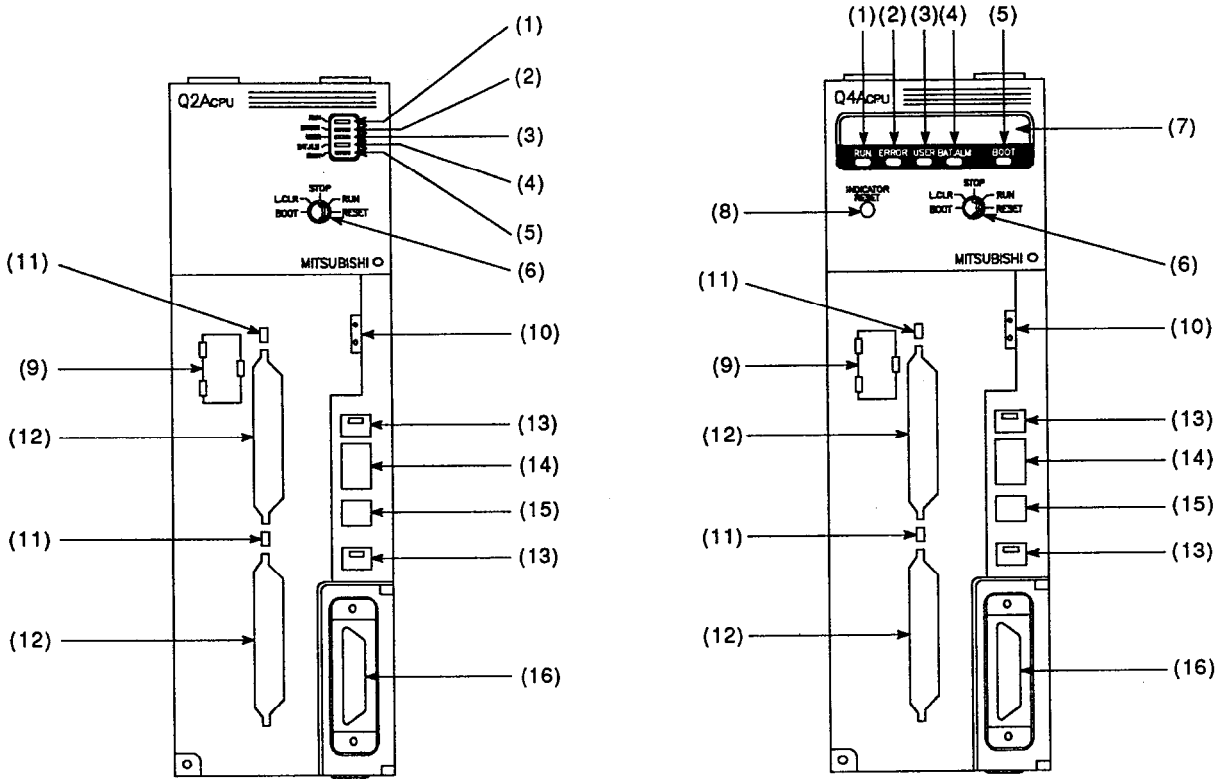
<table>
<tr><th>Item</th><th colspan="5">Specifications</th></tr>
<tr><td>Operating ambient temperature</td><td colspan="5">0 to 55 °C</td></tr>
<tr><td>Storage ambient temperature</td><td colspan="5">–20 to 75 °C</td></tr>
<tr><td>Operating ambient humidity</td><td colspan="5">10 to 90 % RH (no dewing)</td></tr>
<tr><td>Storage ambient humidity</td><td colspan="5">10 to 90 % RH (no dewing)</td></tr>
<tr><td rowspan="8">Vibration resistance</td><td rowspan="8">Conforms to JIS* B 3501, IEC 1131-2</td><td colspan="4">Intermittent vibration</td></tr>
<tr><td>Frequency</td><td>Acceleration</td><td>Amplitude</td><td>Sweep count</td></tr>
<tr><td>10 to 57 Hz</td><td>——</td><td>0.075 mm</td><td rowspan="5">10 times in each of X, Y and Z directions (for 80 minutes)</td></tr>
<tr><td>57 to 150 Hz</td><td>9.8 m/s^2{1 G}</td><td>——</td></tr>
<tr><td colspan="3">Continuous vibration</td></tr>
<tr><td>Frequency</td><td>Acceleration</td><td>Amplitude</td></tr>
<tr><td>10 to 57 Hz</td><td>——</td><td>0.035 mm</td></tr>
<tr><td>57 to 150 Hz</td><td>4.9 m/s^2{0.5 G}</td><td>——</td></tr>
<tr><td>Shock resistance</td><td colspan="5">Conforms to JIS* B 3501, IEC 1131-2 (147 m/s^2 {15 G}, 3 times in each of 3 directions)</td></tr>
<tr><td>Operating atmosphere</td><td colspan="5">Free of corrosive gases</td></tr>
<tr><td>Operating altitude</td><td colspan="5">No greater than 2000 m</td></tr>
<tr><td>Installation site</td><td colspan="5">Control panel</td></tr>
<tr><td>Overvoltage category</td><td colspan="5">11 or lower</td></tr>
<tr><td>Contamination level</td><td colspan="5">2 or lower</td></tr>
</table>

REMARK

JIS* : Japanese Industrial Standard

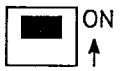
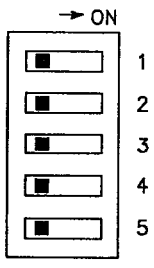
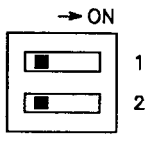
15.2 Part Identification and Setting

The names of module parts and their settings are described here.



* Seen with front cover removed

No.	Name	Application
(1)	RUN LED	Indicates the operating state of the CPU. ON : A sequence program is being executed with the RUN/STOP key switch in the RUN or STEP RUN position. OFF : Operation is stopped, with the RUN/STOP key switch in the STOP, PAUSE, or STEP RUN position. Or, an error that stops operation has been detected. Flashing: The key switch has been turned from STOP to RUN after writing a program in the STOP state. The CPU is not in the RUN state. To set it to the RUN state, turn the key switch from RUN to STOP and back to RUN. Alternatively, perform a reset operation with the key switch. (With Q3ACPU and Q4ACPU, the message "PRG.CHECK!!" is displayed on the LED indicator.)
(2)	ERROR LED	ON : A self-diagnosis error that does not stop operation — other than a battery error — has been detected. (When the parameter setting is made for operation to continue when the error occurs.) OFF : Normal Flashing: An error that stops operation has been detected.
(3)	USER LED	ON : An error has been detected by means of the CHK instruction, or an annunciator, F, has come ON. (With Q3ACPU or Q4ACPU, a message or the comment for the annunciator is displayed on the LED indicator.) OFF : Normal Flashing: When latch clear is performed. (With Q3ACPU or Q4ACPU, the message "L.CLR RDY" is displayed on the LED indicator.)

<table>
<tr><th>No.</th><th>Name</th><th>Application</th></tr>
<tr><td>(4)</td><td>BAT. ALARM LED</td><td>ON : A battery alarm has occurred due to low voltage of the CPU module battery or memory card battery.
OFF : Normal</td></tr>
<tr><td>(5)</td><td>BOOT LED</td><td>ON : When execution of the boot operation is completed.
OFF : When no boot operation has been executed.</td></tr>
<tr><td>(6)</td><td>RUN/STOP key switch</td><td>RUN/STOP: To start/stop running a sequence program.
L. CLR : To clear (turn OFF, or clear to "0") all data in the latch area set in the parameters.
To clear sampling trace and status latch registrations.
RESET : To reset the hardware. To reset an error occurring during operation and initialize operation.</td></tr>
<tr><td>(7)</td><td>LED indicator (Q3A, Q4ACPU only)</td><td>Can display up to 16 characters.
Displays comments for errors occurring in self-diagnosis, comments in accordance with LED display instructions, clock data in accordance with SET SM212, annunciator F number comments in accordance with SET F instructions.</td></tr>
<tr><td>(8)</td><td>Indicator reset switch (Q3A, Q4ACPU only)</td><td>Switch to clear the LED indicator display; the next data (if there is any) is then displayed.</td></tr>
<tr><td>(9)</td><td>Battery (A6BAT)</td><td>Backup battery for the internal memory and the memory back up function.</td></tr>
<tr><td>(10)</td><td>Battery connector pin</td><td>For connection to the battery lead wire.
(To prevent battery discharge, the battery lead wire is disconnected from the connector on shipping.)</td></tr>
<tr><td>(11)</td><td>Memory card EJECT button</td><td>Used to eject the memory card from the CPU.</td></tr>
<tr><td>(12)</td><td>Memory card installing connector</td><td>Connector for installing the memory card in the CPU.</td></tr>
<tr><td>(13)</td><td>Memory card in/out switch
(with built-in LED)
  </td><td>Used to prohibit/permit installation or removal of a memory card while the power is ON.
Set to OFF on shipping.
ON : Card cannot be removed (LED lit)
OFF : Card can be removed (LED not lit)</td></tr>
<tr><td rowspan="5">(14)</td><td rowspan="5">System setting switch 1
  </td><td>Sets settings required to operate the CPU. All switches are set to OFF on shipping.</td></tr>
<tr><td>SW1: Boot setting. Sets the memory used for operation.
ON : Boot operation
OFF: Internal memory operation</td></tr>
<tr><td>SW2 to 4: Parameter area. Sets the memory in which the parameters are written.
<table>
<tr><th rowspan="2"></th><th rowspan="2">Internal Memory</th><th colspan="2">Memory Card A</th><th colspan="2">Memory Card B</th></tr>
<tr><th>RAM</th><th>ROM</th><th>RAM</th><th>ROM</th></tr>
<tr><td>SW2</td><td>OFF</td><td>ON</td><td>OFF</td><td>ON</td><td>OFF</td></tr>
<tr><td>SW3</td><td>OFF</td><td>OFF</td><td>ON</td><td>ON</td><td>OFF</td></tr>
<tr><td>SW4</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td><td>ON</td></tr>
</table></td></tr>
<tr><td>SW5: System protect. Batch prohibition of data writing and control directions with respect to the CPU.
ON : System protection ON
OFF: System protection OFF</td></tr>
<tr><td></td></tr>
<tr><td rowspan="3">(15)</td><td rowspan="3">System setting switch 2
  </td><td>Sets settings required to operate the CPU. All switches are set to OFF on shipping.</td></tr>
<tr><td>SW1: For future expansion. No function at present.</td></tr>
<tr><td>SW2: Peripheral protocol. Selects the type of peripheral device connected to the peripheral interface of the CPU.
(When accessing an ACPU in another station from a peripheral device for ACPU, set this switch to "ON". The setting becomes valid immediately on switching.)
ON : Peripheral device for ACPU
OFF: Peripheral device for QnACPU</td></tr>
<tr><td>(16)</td><td>RS-422 connector</td><td>Connector for connecting to a peripheral device.</td></tr>
</table>

15.3 Relationship Between Switch Operations and LEDs/LED Indicator

(1) Program writing with CPU in the STOP state

To write a program to the CPU while it is in the STOP state, use the following procedure.

(a) Set the key switch to STOP

RUN LED : OFF .... CPU in STOP state
Q3ACPU, Q4ACPU indicator : OFF (program write executed)

(b) Set the key switch to RUN

RUN LED : OFF .... CPU in RESET state
Q3ACPU, Q4ACPU indicator : OFF

(c) Set the key switch to RUN

RUN LED : ON ..... CPU in RUN state
Q3ACPU, Q4ACPU indicator : OFF

POINTS

- When using a QnACPU, after writing a program (excluding write during RUN), perform a reset operation before setting the CPU to RUN.
- If the key switch is set to RUN without first performing a reset operation, the CPU will give the following indication and will remain the STOP state.

  RUN LED : Flashes
  Q3ACPU, Q4ACPU indicator : Displays "PRG.CHECK!!"

  If this happens, the CPU can be set to the RUN state by performing a reset operation with the key switch.
  However, information inside the CPU, such as device data, is cleared.
- Another method that avoids clearing data is to turn the key switch to STOP and then RUN again.
  However, in this case note that pulse instructions may not operate correctly.

(2) Performing latch clear

To perform a latch clear operation, operate the RUN/STOP key switch as follows.

(a) Turn the key switch to L.CLR three times.

USR LED	:	Flashes	Ready for latch clear execution
Q3ACPU, Q4ACPU indicator	:	Displays "L.CLR RDY"	

(b) Turn the key switch to "L.CLR" one more time.

USR LED	:	ON for 2 seconds..	Latch clear completed
Q3ACPU, Q4ACPU indicator	:	"L.CLR OK" displayed for 2 seconds	

POINTS

- It is possible to set whether the latch clear function is effective or not for each device by device setting in the parameter mode.
- Apart from using the key switch, latch clear can also be executed by using GPPQ (see Section 10.6.5).

(3) Removing a memory card with power ON

To remove a memory card while the power is ON, operate the memory card in/out switch as follows.

(a) In/out switch: ON

LED in the switch : ON ..... Memory card removal prohibited

(b) In/out switch: OFF

LED in the switch : OFF .... Memory card removal permitted (remove the memory card)

POINTS

- The LED in the in/out switch may not come ON if the memory card is being used for CPU system functions (sampling trace, status latch, etc.) or by a program.
  In this case, abort the function or program that is using the memory card.
  After aborting it, confirm that the LED in the in/out switch has gone OFF, then remove the memory card.
- After removing the memory card, do not turn the memory card in/out switch ON. This will cause an error.

(4) Installing a memory card with the power ON

To install a memory card while the power is ON, operate the memory card in/out switch as follows.

(a) Install the memory card

(b) In/out switch: ON

LED in the switch : ON . . . . . Memory card removal prohibited

POINTS

- After installing a memory card, set the memory card in/out switch to ON. If it is not set to ON it will not be possible to use the memory card.
- Note that the scan time will be lengthened by up to 10 ms after installation of a memory card because mount processing is performed again.

16. POWER SUPPLY MODULE

This section describes the specifications and selection of power supply modules.

16.1 Specifications

16.1.1 Power supply module specifications

The specifications of power supply modules are shown below.

Power Supply Module Specifications

Item		Specifications							
		A61P	A61PEU	A62P	A62PEU	A63P	A65P	A66P	A67P
Base unit loading position		Power supply module loading slot						I/O module loading slot	Power supply module loading slot
Input voltage		100–120 VAC$^{+10\ \%}_{-15\ \%}$ (85 to 132 VAC)				24 VDC$^{+30\ \%}_{-35\ \%}$ (15.6 to 31.2 VDC)	100–120 VAC$^{+10\ \%}_{-15\ \%}$ (85 to 132 VAC)		110 VDC (85 to 140 VDC)
		200–240 VAC$^{+10\ \%}_{-15\ \%}$ (170 to 264 VAC)					200–240 VAC$^{+10\ \%}_{-15\ \%}$ (170 to 264 VAC)		
Input frequency		50/60 HZ±5 %				—	50/60 HZ±5 %		—
Max. input apparent power		110 VA				65 W	110 VA	95 VA	65 W
Inrush current		20 A, within 8 ms				100 A, within 1 ms	20 A, within 8 ms		
Rated output current	5 VDC	8 A		5 A		8 A	2 A	—	8 A
	24 VDC	—		0.8 A		—	1.5 A	1.2 A	—
*1 Overcurrent protection	5 VDC	8.8 A or higher		5.5 A or higher		8.5 A or higher	2.2 A or higher	—	8.5 A or higher
	5 VDC	—		1.2 A or higher		—	2.3 A or higher	1.7 A or higher	—
*2 Overvoltage protection	5 VDC	5.5 to 6.5 V		5.5 to 6.5 V		5.5 to 6.5 V	5.5 to 6.5 V	—	5.5 to 6.5 V
	24 VDC	—							
Efficiency		65 % or higher							
Power indicator		Power LED display							
Terminal screw size		M4 x 0.7 x 6						M3 x 0.5 x 6	M4 x 0.7 x 6
Applicable wire size		0.75 to 2 mm^2							
Applicable solderless terminal		Same as *a	RAV 1.25–4, RAV2–4	Same as *a	RAV 1.25–4, RAV2–4	V1.25–4, V1.25–YS4A, V2–S4, V2–YS4A *a		V1.25–3, V1.25–YS3A V2–S3, V2–YS3A	V1.25–4, V1.25–YS4A V2–S4, V2–YS4A
Applicable tightening torque: N·cm [kg·cm] (ib·in)		118 [12] (10)						69 [7] (6)	118 [12] (10)

Item		Specifications							
		A61P	A61PEU	A62P	A62PEU	A63P	A65P	A66P	A67P
External dimensions mm (in)		250 x 55 x 121 (9.8 x 2.1 x 4.7)						250 x 37.5 x 121 (9.8 x 1.5 x 4.7)	250 x 55 x 121 (9.8 x 2.1 x 4.7)
Weight kg (lb)		0.98 (2.16)	0.8 (1.76)	0.94 (2.07)	0.9 (1.98)	0.8 (1.76)	0.94 (2.07)	0.75 (1.65)	0.8 (1.76)
*3 Allowable momentary power interruption time		Less than 20 ms				Less than 1 ms	Less than 20 ms	——	Less than 20 ms (at 100 VDC)
*4 Noise durability		Noise voltage 1500 V.P.P.				Noise voltage 500 V.P.P.	Noise voltage 1500 V.P.P.		Noise voltage 500 V.P.P
Withstanding voltage		Same as *b	——	Same as *b	¤¤	1500 VAC for 1 minute between all AC external terminals together and ground 500 VAC for 1 minute between all DC external terminals together and ground *b			
Insulation resistance		Same as *c	——	Same as *c	¤¤	10 MW or higher, measured with a 500 VDC insulation resistance tester *c			
Insuiation withstand voltage	Between primary and 5 VDC	——	2830 VAC	¤¤	2830 VAC	——			
	Between primary and 24 VDC	——	¤¤	——	2830 VAC	——			

REMARK

The A66P module has the number of occupied slots shown below. 1 slot

POINTS

*1: Overcurrent protection

(a) The overcurrent protection device shuts off the 5 VDC, 24 VDC ladder and stops the system if the current flowing in the ladder exceeds the specified value.
When this device is activated, the power supply module LED is switched off or dimly lit.

(b) If this happens, eliminate the cause of the overcurrent — for example insufficient current capacity or short ladder — then start up the system.
When the current has returned to normal, the system undergoes an initial start.

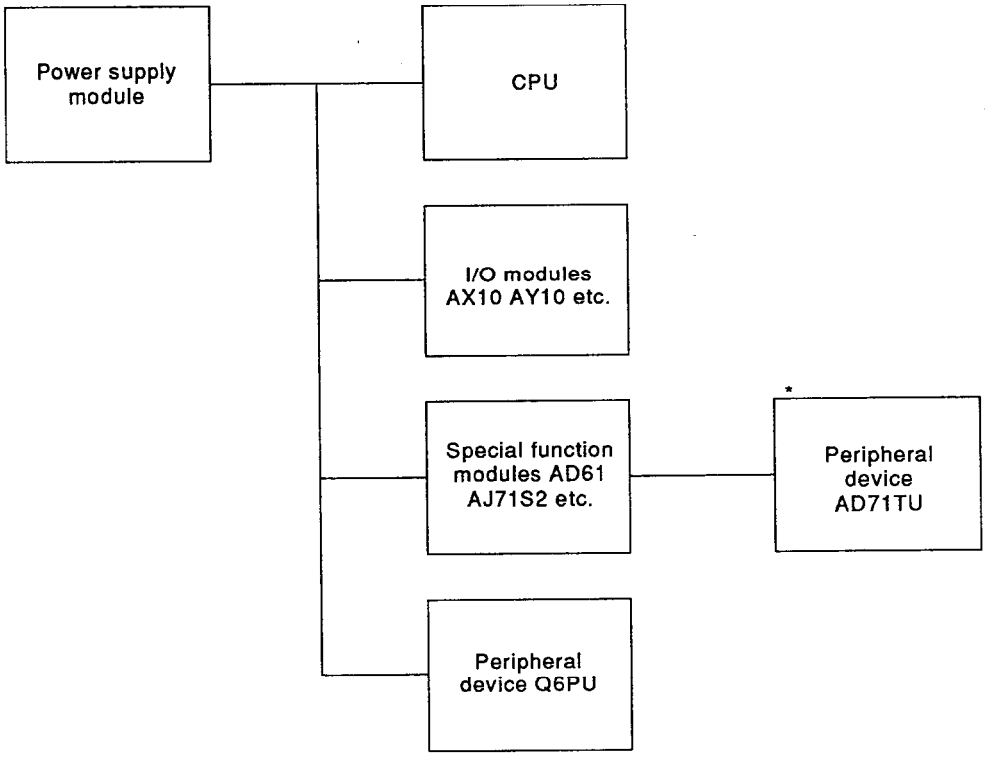
*2: Overvoltage protection
The overvoltage protection device shuts off the 5 VDC ladder and stops the system if an excessive voltage in the range 5.5 to 6.5 V is applied to this ladder.
When this device is activated, the power supply module LED is switched off. If this happens, switch the input power OFF, then back ON to restart the system.
If the system is not booted and the LED remains off, the power supply module must be changed.

*3: Allowable momentary power interruption time
The PC CPU allowable momentary power interruption time varies according to the type of power supply module.
In the case of the A63P module, the allowable momentary power interruption time is defined as from when the 24 VDC stabilized primary supply is cut off until the 24 VDC voltage drops to the defined voltage (15.6 VDC).

*4: Noise durability
Apply the specified noise voltage with a noise simulator with a noise width of 1 μs and a noise frequency of 25 to 60 Hz.

16.1.2 Selection of power supply module

Select the power supply module according to the total current consumption of the I/O modules, special function modules, and peripheral devices supplied by that power supply module. When an A52B, A55B, A58B extension base unit is used, power is supplied from the power supply module of the main base unit: this should be taken into consideration.
Refer to Section 3.3 for the 5 VDC current consumption of I/O modules, special function modules, and peripheral devices.



* When selecting the power supply module, take account of the current consumption of the peripheral devices connected to special function modules.
For example, when an AD71TU is connected to an AD71S2, the current consumption of the AD71TU should also be taken into consideration.

(1) Selection of power supply module when an A52B, A55B, A58B extension base unit is used.

When an A52B, A55B, A58B extension base unit is used, the 5 VDC power is supplied from the power supply module of the main base unit via the extension cable. Therefore, when using an A52B, A55B, A58B, note the following points:

(a) Select the 5 VDC capacity of power supply module in the main base unit so that it can cover the current consumption of 5 VDC used for the A52B, A55B, A58B.

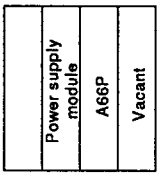
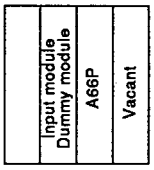
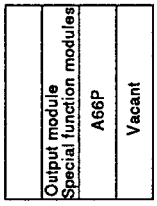
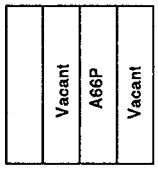
Example: Assuming that the 5 VDC current consumption of main base unit is 5 A and the 5 VDC current consumption of A55B is 2 A, the A61P/A61PEU (5 VDC, 8 A) must be selected as the power supply module on the main base unit.

(b) Since the power is supplied to the A52B, A55B or A58B via the extension cable, some voltage drop occurs in the cable. It is necessary to select the power supply module and cable length to ensure supply of 4.75 VDC or more at the receiving end.
For details of voltage drop, etc., see Section 17.3 "Application Standards of Extension Base Units (A52B, A55B, A58B)"

(2) Notes on use of the A66P

(a) Mount an A66P either in the slot of the base unit that is furthest to the right, or in a slot with a vacant slot, dummy slot, or blank cover to its right.

(b) The A66P output current (24 VDC) depends on the left-hand adjacent module, as follows.

Left Hand Adjacent Module	Power Supply Module	Input Module Dummy Module	Output Module Special Function Modules	Vacant
Configuration				
Max. output current for 24 VDC	0.5 A	1.2 A	1.0 A	1.5 A

16.1.3 Fuse specifications

This section describes the specifications of fuses used for the power supply modules and output modules.

Fuse Specifications

Item \ Model Name	GTH4	SM6.3A	MF51NM8	HP-32	HP-70K	MP-20	MP-32	MP-50
Application	Power supply module For A61P, A61PEU, A62P, A62PEU, A65P, A66P, A67P	Power supply module For A63P	Output module For AY11E, AY13E	Output module For AY23	Output module For AY22	Output module For AY50, AY80	Output module For AY60	Output module For AY60E
Type	Cartridge type	Cartridge type	Cartridge type	Plug type	Plug type	Plug type	Plug type	Plug type
Rated current	4 A	6.3 A	8 A	3.2 A	7 A	2 A	3.2 A	5 A
External dimensions mm (inch)	φ6 x 32 (0.2 x 1.2)	φ6 x 32 (0.2 x 1.2)	φ5.2 x 20 (0.2 x 0.8)	30.3 x 8 x 20 (1.2 x 0.3 x 0.8)	30.3 x 8 x 20 (1.2 x 0.3 x 0.8)	17.2 x 5.5 x 19 (0.7 x 0.2 x 0.7)	17.2 x 5.5 x 19 (0.7 x 0.2 x 0.7)	17.2 x 5.5 x 19 (0.7 x 0.2 x 0.7)

16.2 Handling

16.2.1 Handling instructions

Things to note during opening the power supply module package through mounting the power supply module are given below.

(1) The case, terminal connector and pin connector of the power supply module are made of resin: do not drop the module or subject it to strong impacts.

(2) Do not remove printed circuit boards from the housing. This can cause failure.

(3) Ensure that no conductive debris can enter the module. If any does, make sure that it is removed. Guard particularly against wire offcuts.

(4) Tighten the module mounting and terminal screws as specified below.

Screw		Tightening Torque N·cm [kg·cm] (lb·in)
Module terminal block installation screws	(M3)	49 to 78 [5 to 8] (4.3 to 6.8)
Module terminal block installation screws	(M4)	98 to 137 [10 to 14] (8.6 to 12)
Module mounting screws (optional)	(M4)	78 to 118 [8 to 12] (6.8 to 10.4)

(5) To load the module onto the base unit, hook the two lower lugs into the cut out and gently swing the module into place. Ensure that the top catch engages. To remove, press the top catch and swing the module out before unhooking the lower lugs (see Section 19.5).

16.2.2 Part identification and setting

The names and descriptions of each of the parts of the power supply modules are given below.

(1) Names and description of parts of the A61P module

Module fixing hook
Hook for fixing the module to the base unit.

"POWER" LED
LED for indicating 5 VDC power

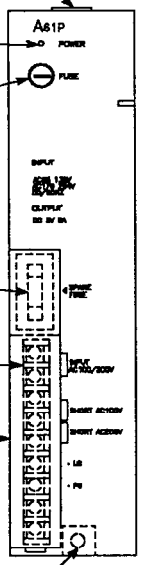
Power fuse, fuse holder
4 A cartridge fuse for AC input power is secured by the fuse holder.

Spare fuse for power supply
Spare fuse for power supply, mounted on rear side of the terminal cover.

Terminal block
For details, see below. (Located under the terminal cover)

Terminal cover
Cover for protection of terminal block. Remove during wiring. Re-install after wiring.

Module fixing screw mounting hole
Allows the module to be secured with a screw in addition to the module fixing hook. (For M4 screw)



Terminal details

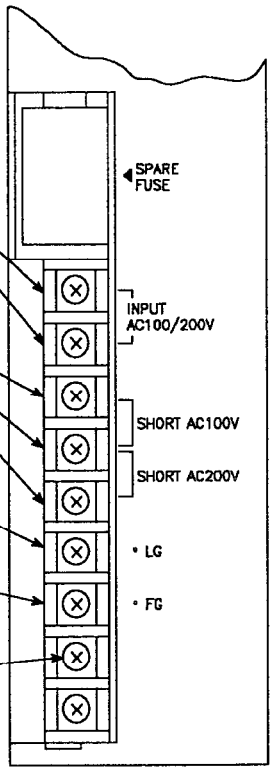
Power input terminals
Power input terminals to which AC power of 100 VAC or 200 VAC

Applied voltage select terminals
Terminals for selecting applied voltage. Use 100 VAC or 200 VAC as described below. When 100 VAC is input, connect together the "SHORT AC100V" terminals with the jumper provided. When 200VAC is input, connect together the "SHORT AC200V" terminals with the jumper provided.

LG terminal
Grounding of power filter. Has half the input potential.

FG terminal
Connection terminal connected to the shielding pattern on printed circuit board.

Terminal screw
M4 x 0.7 x 6



(2) Names and description of parts of the A61PEU module

Module fixing hook
Hook for fixing the module to the base unit.

"POWER" LED
LED for indicating 5 VDC power

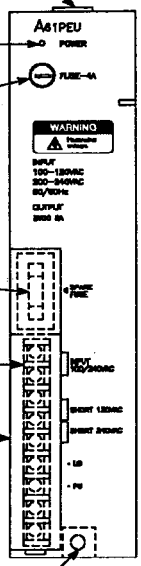
Power fuse, fuse holder
4 A cartridge fuse for AC input power is secured by the fuse holder.

Spare fuse for power supply
Spare fuse for power supply, mounted on rear side of the terminal cover.

Terminal block
For details, see below. (Located under the terminal cover)

Terminal cover
Cover for protection of terminal block. Remove during wiring. Re-install after wiring.

Module fixing screw mounting hole
Allows the module to be secured with a screw in addition to the module fixing hook. (For M4 screw)



Terminal details

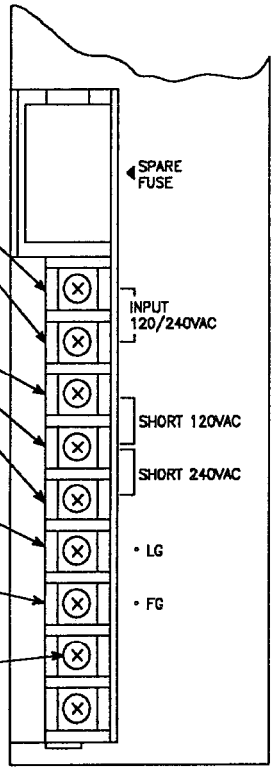
Power input terminals
Power input terminals to which AC power of 120 VAC or 240 VAC

Applied voltage select terminals
Terminals for selecting applied voltage. Use 120 VAC or 240 VAC as described below. When 120 VAC is input, connect together the "SHORT AC120V" terminals with the jumper provided. When 240 VAC is input, connect together the "SHORT AC240V" terminals with the jumper provided.

LG terminal
Grounding of power filter. Has half the input potential.

FG terminal
Connection terminal connected to the shielding pattern on printed circuit board.

Terminal screw
M4 x 0.7 x 6



(3) Names and description of parts of the A62P and A65P modules

Module fixing hook
Hook for fixing the module to the base unit.

"POWER" LED
LED for indicating 5 VDC power

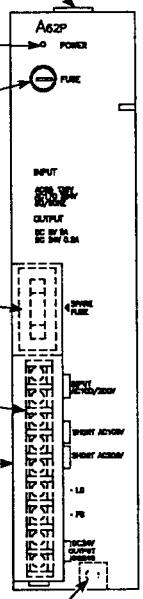
Power fuse, fuse holder
4 A cartridge fuse for AC input power is secured by the fuse holder.

Spare fuse for power supply
Spare fuse for power supply, mounted on rear side of the terminal cover.

Terminal block
For details, see below. (Located under the terminal cover)

Terminal cover
Cover for protection of terminal block. Remove during wiring. Re-install after wiring.

Module fixing screw mounting hole
Allows the module to be secured with a screw in addition to the module fixing hook. (For M4 screw)



Terminal details

Power input terminals
Power input terminals to which AC power of 100 VAC or 200 VAC

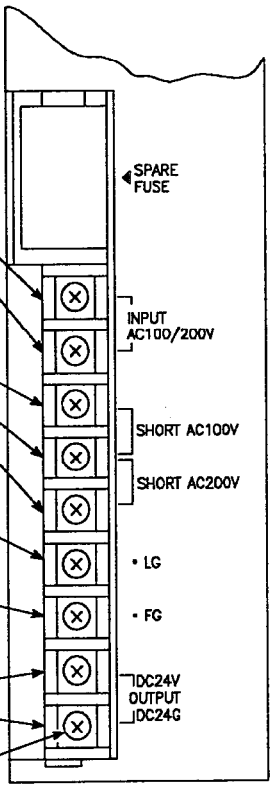
Applied voltage select terminals
Terminals for selecting applied voltage. Use 100 VAC or 200 VAC as described below. When 100 VAC is input, connect together the "SHORT AC100V" terminals with the jumper provided. When 200 VAC is input, connect together the "SHORT AC200V" terminals with the jumper provided.

LG terminal
Grounding of power filter. Has half the input potential.

FG terminal
Connection terminal connected to the shielding pattern on printed circuit board.

24 VDC, 24 GDC terminals
For supply to output module which requires 24 V inside the module. (Supplied to the module via external wiring)

Terminal screw
M4 x 0.7 x 6



(4) Names and description of parts of the A62PEU modules

Module fixing hook
Hook for fixing the module to the base unit.

"POWER" LED
LED for indicating 5 VDC power

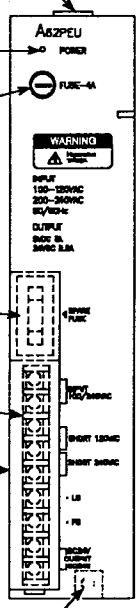
Power fuse, fuse holder
4 A cartridge fuse for AC input power is secured by the fuse holder.

Spare fuse for power supply
Spare fuse for power supply, mounted on rear side of the terminal cover.

Terminal block
For details, see below. (Located under the terminal cover)

Terminal cover
Cover for protection of terminal block. Remove during wiring. Re-install after wiring.

Module fixing screw mounting hole
Allows the module to be secured with a screw in addition to the module fixing hook. (For M4 screw)



Terminal details

Power input terminals
Power input terminals to which AC power of 120 VAC or 240 VAC

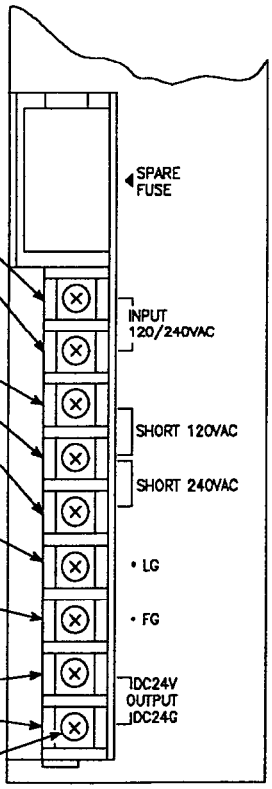
Applied voltage select terminals
Terminals for selecting applied voltage. Use 120 VAC or 240 VAC as described below. When 120 VAC is input, connect together the "SHORT AC120V" terminals with the jumper provided. When 240 VAC is input, connect together the "SHORT AC240V" terminals with the jumper provided.

LG terminal
Grounding of power filter. Has half the input potential.

FG terminal
Connection terminal connected to the shielding pattern on printed circuit board.

24 VDC, 24 GDC terminals
For supply to output module which requires 24 V inside the module. (Supplied to the module via external wiring)

Terminal screw
M4 x 0.7 x 6



(5) Names and description of parts of the A63P and A67P modules

Module fixing hook
Hook for fixing the module to the base unit.

"POWER" LED
LED for indicating 5 VDC power.

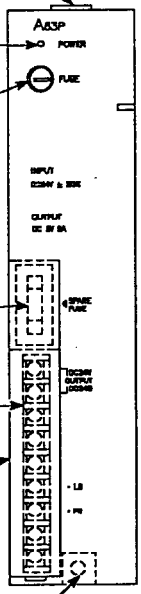
Power fuse, fuse holder
Cartridge fuse for DC input power is fixed by the fuse holder. The rating for the fuses are as follows.
A63P: 6.3 A (SM6.3 A)
A67P: 4 A (GTH4)

Spare fuse for power supply
Spare fuse for power supply, mounted on rear side of the terminal cover.

Terminal block
For details, see below. (Located under the terminal cover).

Terminal cover
Cover for protection of terminal block. Remove during wiring. Re-install after wiring.

Module fixing screw mounting hole
Allows the module to be secured with a screw in addition to the module fixing hook. (For M4 screw).



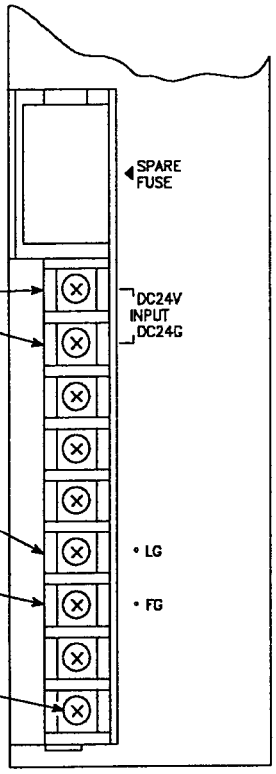
Terminal details

Power input terminals
Power input terminals for A63P: 24 VDC, A67P: 100 VDC.
The power fuse will be blown if the 24 VDC connection is made with the wrong polarity.

LG terminal
Grounding of power filter.

FG terminal
Connection terminal connected to the shielding pattern on printed circuit board.

Terminal screw
M4 x 0.7 x 6



(6) Names and description of parts of the A66P module

Module fixing hook
Hook for fixing the module to the base unit.

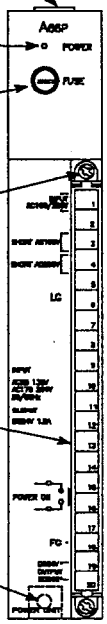
"POWER" LED
LED for indicating 5 VDC power.

Power fuse, fuse holder
4 A cartridge fuse for AC input power is secured by the fuse holder.

Terminal block fixing screw
Screw for installing and fixing the terminal block to the module.

Terminal block
For details, see below. (Located under the terminal cover).

Module fixing screw mounting hole
Allows the module to be secured with a screw in addition to the module fixing hook. (For M4 screw).



Terminal details

Power input terminals
Power input terminals to which AC power of 100 VAC or 200 VAC

Applied voltage select terminals
Terminals for selecting applied voltage. Use 100 VAC or 200 VAC as described below. When 100 VAC is input, connect together the "SHORT AC100V" terminals with the jumper provided. When 200 VAC is input, connect together the "SHORT AC200V" terminals with the jumper provided.

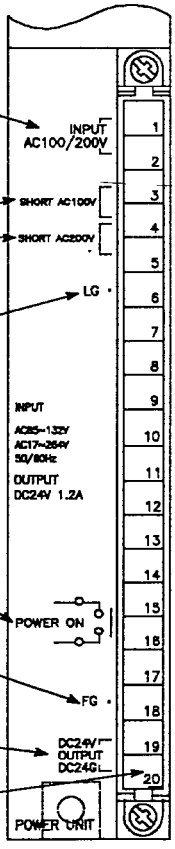
LG terminal
Grounding of power filter. Has half the input potential.

Power ON terminal
Contact terminal which conducts if the 24 VDC output is normal when power input turns on.

FG terminal
Connection terminal connected to the shielding pattern on printed circuit board

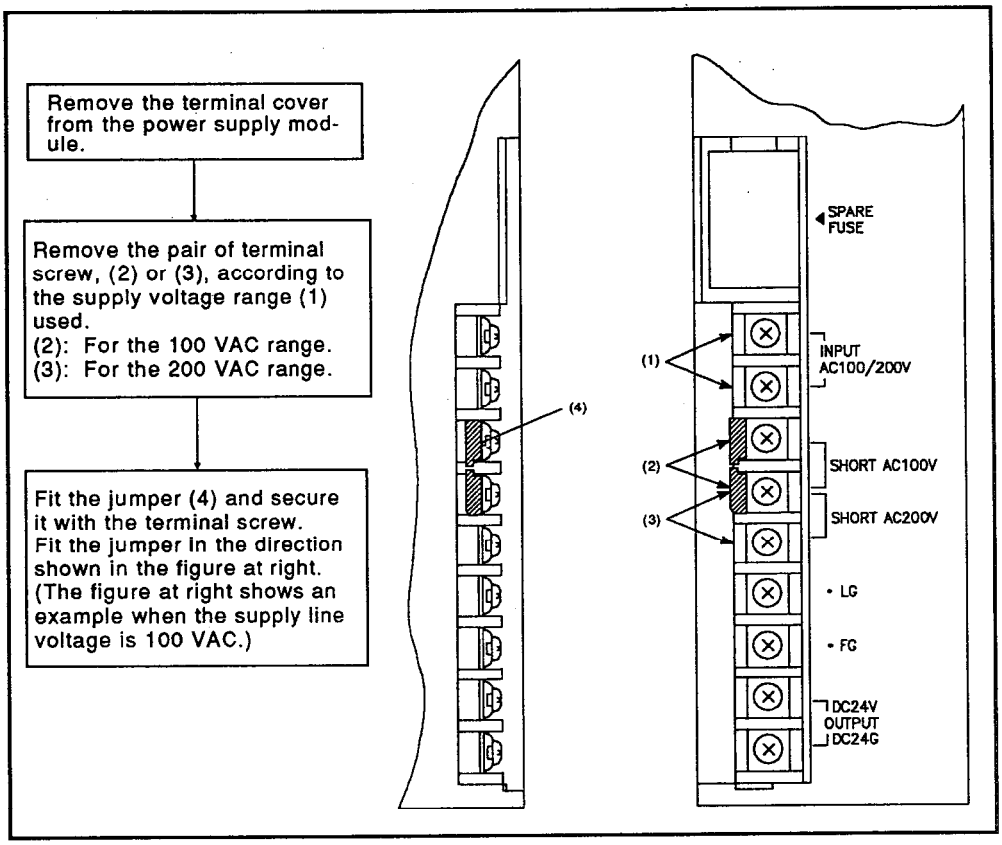
24 VDC, 24 GDC terminals
For supply to output module which requires 24 V inside the module. (Supplied to the module via external wiring)

Terminal screw
M3 x 0.5 x 6



(7) Setting

For A61P(EU), A62P(EU), A65P or A66P, the input voltage range, 100V or 200 V, must be specified by placing a jumper (supplied) across two terminals as described below:



POINT

If the setting differs from the supply line voltage, the following occurs: do not mis-set.

	Supply Line Voltage	
	100 VAC	200 VAC
Setting to 100 VAC (jumper fitted as indicated at (2))	—	The power supply module is damaged. (The CPU is not damaged.)
Setting to 200 VAC (jumper fitted as indicated at (3))	No error occurs in the module. However, the CPU does not operate.	—
No setting (jumper not fitted)	No error occurs in the module. However, the CPU does not operate.	

17. BASE UNITS AND EXTENSION CABLES

This section describes the specifications of the base units (main base units and extension base units) and extension cables that can be used with the system.

17.1 Specifications of Base Units

(1) Specifications of main base units

Item \ Model Name	A32B	A32B-S1	A35B	A38B/A38HB
Loaded I/O modules	2 can be loaded		5 can be loaded	8 can be loaded
Extension connection	Impossible	Possible	Possible	Possible
Mounting hole size	ϕ6 mm (0.24 in) dia. pear-shaped hole (for M5 screw)			
External dimensions mm (in)	247 x 250 x 29 (9.6 x 9.8 x 1.1)	268 x 250 x 29 (10.5 x 9.8 x 1.1)	382 x 250 x 29 (14.9 x 9.8 x 1.1)	480 x 250 x 29 (18.7 x 9.8 x 1.1)
Weight kg (lb)	0.96 (2.1)	1.3 (2.9)	1.5 (3.3)	1.9 (4.2)

(2) Specifications of extension base units

Item \ Model Name	A62B	A65B	A68B	A52B	A55B	A58B
Loaded I/O modules	2 can be loaded	5 can be loaded	8 can be loaded	2 can be loaded	5 can be loaded	8 can be loaded
Power supply module loading	Power supply module required			Not required. (See POINTS below.)		
Mounting hole size	ϕ6 mm (0.24 in) dia. pear-shaped hole (for M5 screw)			ϕ6 mm (0.24 in) dia. pear-shaped hole (for M5 screw)		
Terminal screw size	—			M4 x 0.7 x 6 (FG terminal)		
Applicable wire size	—			0.75 to 2 mm^2		
Applicable solderless terminal	—			(V)1.25–4, (V)1.25–YS4, (V)2–YS4A Applicable tightening torque: 118N·cm(12kg·cm) [10 lb·in]		
External dimensions mm (in)	283 x 250 x 29 (11.0 x 9.8 x 1.1)	352 x 250 x 29 (13.7 x 9.8 x 1.1)	466 x 250 x 29 (18.2 x 9.8 x 1.1)	183 x 250 x 29 (7.1 x 9.8 x 1.1)	297 x 250 x 29 (11.6 x 9.8 x 1.1)	411 x 250 x 29 (16.0 x 9.8 x 1.1)
Weight kg (lb)	1.1 (2.4)	1.4 (3.1)	1.9 (4.2)	1.0 (2.2)	1.2 (2.6)	1.7 (3.7)

*1: For the installation of the dustproof cover, see Section 19.6.

POINTS

(1) The 5 VDC power supply of the A52B, A55B, and A58B must be supplied from a power supply module mounted on the main base unit.

(2) When using an A52B, A55B or A58B base unit, refer to Section 16.1.2 "Selection of Power Supply Modules" and Section 17.3 "Application Standards of Extension Base Units (A52B, A55B, and A58B)".

17.1.1 Main base unit for high-speed access (A38HB)

The high-speed access main base unit (A38HB) is a base unit designed to make accessing of the buffer memories of special function modules from a QnACPU faster.

POINTS

(1) The A38HB can only perform high-speed access with respect to the buffer memories of special function modules.
The I/O devices of I/O modules are not accessed at high speed but at the same access speed as a conventional main base unit.

(2) When an extension base unit is connected to an A38HB, the buffer memories of the special function modules on the extension base unit are not accessed at high speed. They are accessed at the same speed as they would be if the extension base unit were connected to a conventional main base unit.

CAUTION

The A38HB base unit is dedicated to QnACPU and cannot be used with ACPU.

17.2 Specifications of Extension Cables

Item \ Model Name	AC06B	AC12B	AC30B
Cable length m (ft)	0.6 (2.0)	1.2 (3.9)	3 (9.8)
Resistance value of 5 VDC supply line (Ω (at 55 °C))	0.019	0.028	0.052
Application	• For connection between main base and extension base • For connection between extension bases		
Weight kg (lb)	0.34 (0.75)	0.52 (1.14)	1.06 (2.33)

17.3 Application Standards of Extension Base Units (A52B, A55B, and A58B)

With the A52B, A55B and A58B extension base units, 5 VDC is supplied from the power supply module of the main base unit via the extension cable. (Power supply is not available from the power supply modules used for the A62B, A65B and A68B units.)
Since some voltage drop occurs within the extension cable, the specified voltage may not supplied to the receiving end, resulting in erroneous inputs and outputs.
It is recommended that the A52B, A55B and A58B units be connected after the main base unit to minimize voltage drop.
Determine applicability of the A52B, A55B and A58B units using the voltage calculation method described below.

(1) Selection conditions

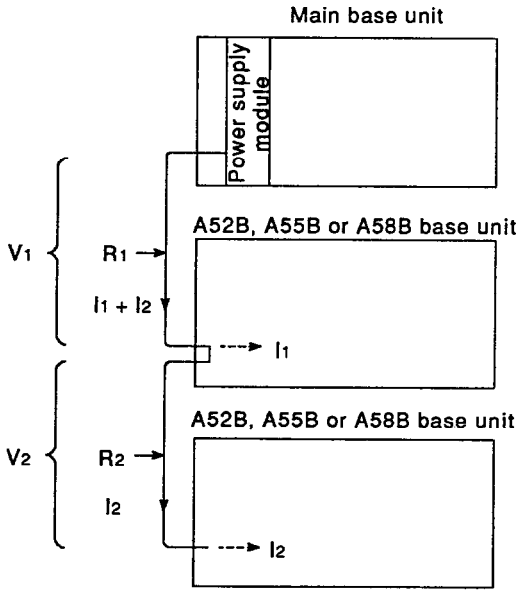
(a) 4.75 VDC or more should be supplied at the farthest base unit from the main base unit.

(2) Calculation of voltage at farthest unit

(a) The 5 VDC output voltage of power supply module fluctuates by approx. 0.1 V. Hence in the worst case the supply voltage will drop to 4.9 V.

(b) Resistance value of cable

AC06B..... 0.019 Ω
AC12B..... 0.028 Ω
AC30B..... 0.052 Ω



V1: Voltage drop in cable between main base unit and extension base unit
V2: Voltage drop in cable between extension base units
R1: Resistance of cable between main base unit and extension base unit
R2: Resistance of cable between extension base units
I1: Current consumption of 5 VDC used for the 1st extension stage
I2: Current consumption of 5 VDC used for the 2nd extension stage

Voltage drops V1 and V2 are:
$V_1 = R_1 (I_1 + I_2)$
$V_2 = R_2 i_2$

The voltage of the receiving end at the 2nd extension stage must satisfy the following expression:

Farthest base unit voltage = $4.9 - (V_1 + V_2) > 4.75$

In order to satisfy the condition that the voltage at the farthest base unit should be 4.75 V or more, the following conditions should hold

$4.9 - 4.75 \geq V_1 + V_2$

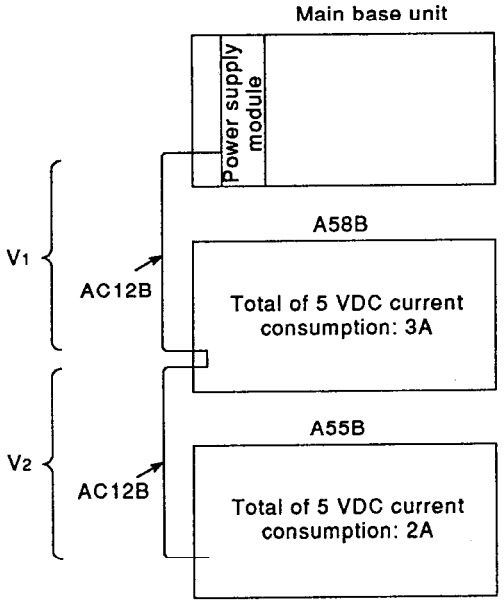
$0.15 \geq R_1 (I_1 + I_2) + R_2 I_2$

Under the above conditions, the A52B, A55B or A58B can be used for two extension stages.

Therefore, the number of A52B, A55B or A58B extension base units which may be powered from one power supply is limited by the voltage drop between the main base unit and the farthest base unit.

$0.15 \geq$ total of voltage drop up to receiving end

(3) Calculation example



Resistance value of AC12B = 0.028 Ω

$V_1 = 0.028 \times (3 + 2)$
$= 0.14$ [V]

$V_2 = 0.028 \times 2$
$= 0.056$ [V]

$V_1 + V_2 = 0.196$ [V]

(Receiving end voltage of 2nd extension stage is 4.9 − 0.196 = 4.704 V)

Therefore, since the voltage drop is higher than 0.15 V, the A55B cannot be used at the 2nd extension stage under this condition. In this case, the A55B can be used by changing the cable to AC06B (resistance value = 0.019 Ω)

$V_1 = 0.019 \times (3 + 2)$
$= 0.095$ [V]

$V_2 = 0.019 \times 2$
$= 0.038$ [V]

$V_1 + V_2 = 0.133$ [V]

(Receiving end voltage of 2nd extension stage is 4.9 − 0.133 = 4.767 V)

Thus, since the voltage drop is less than 0.15 V, the A55B can be used for the 2nd extension stage under this condition.

REMARK

Voltage drop when the A62B, A65B, and A68B units are connected between the main base unit and the A52B, A55B, and A58B units should be calculated following the method indicated below and referring to the method described above.

- Obtain the total of resistance of the extension cables between the main base unit and the A52B, A55B, and A58B units via the A62B, A65B, and A68B units, and also obtain the current consumption of the 5 VDC voltage of the A52B, A55B, and A58B which flows in the extension cables. Use these values for calculation of voltage drop.
  (Current consumption at the A62B, A65B, and A68B units is not necessary for this calculation.)

POINT

When using I/O modules and special function modules whose internal current consumption is large, load these modules in an extension base which requires a power supply module (A62B, A65B, or A68B).

17.4 Handling Instructions

The handling precautions to be taken between unpacking and mounting base units are as follows.

(1) The terminal connector and pin connector of the base unit are made of resin: do not drop the unit or subject it to strong impacts.

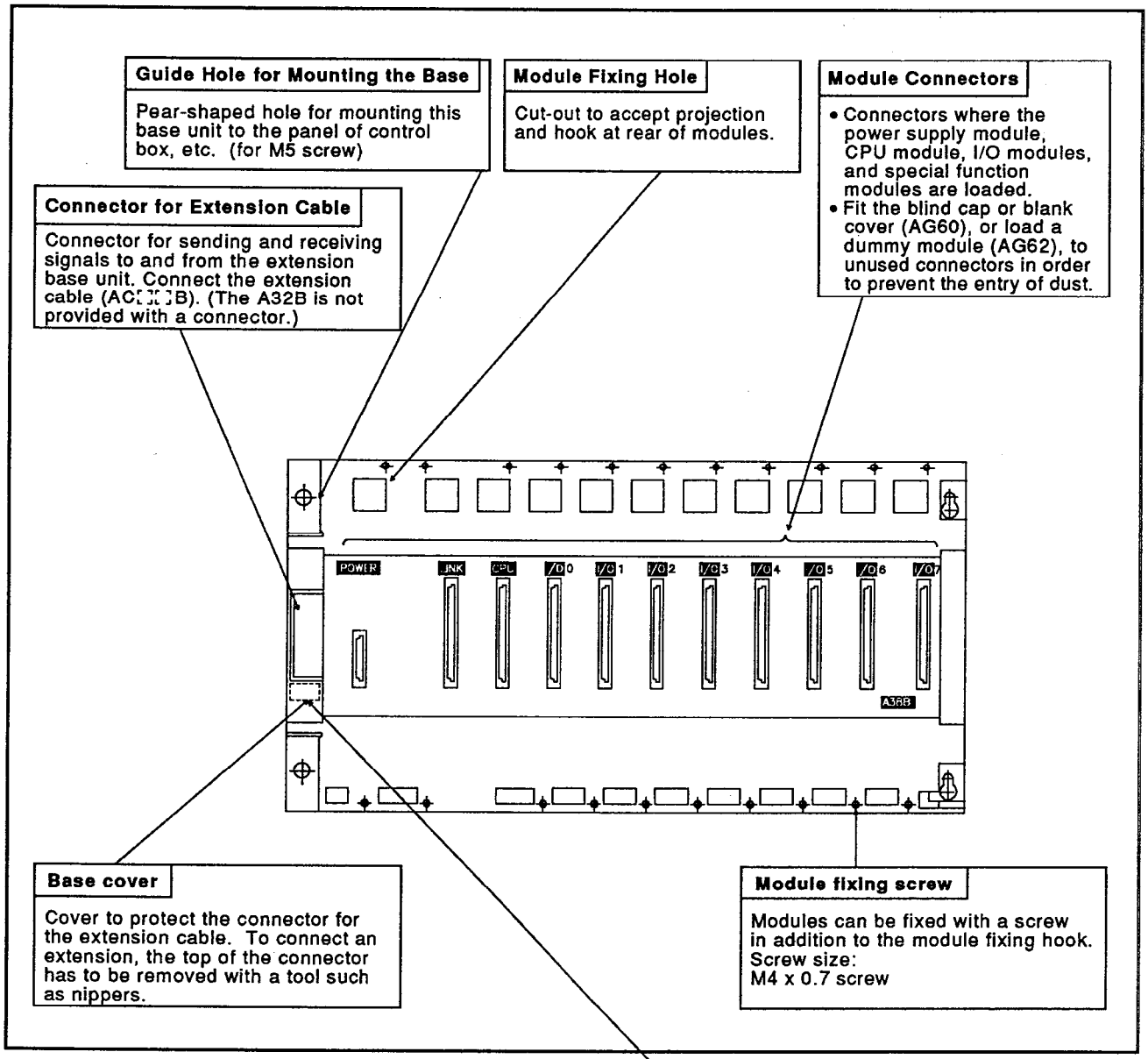
(2) Do not remove printed circuit boards from the housing. There are no user-serviceable parts on the boards.

(3) Ensure that no conductive debris can enter the unit. If any does, make sure that it is removed. Guard particularly against wire offcuts.

17.5 Part Identification

The names and descriptions of each of the parts of base units are given below.

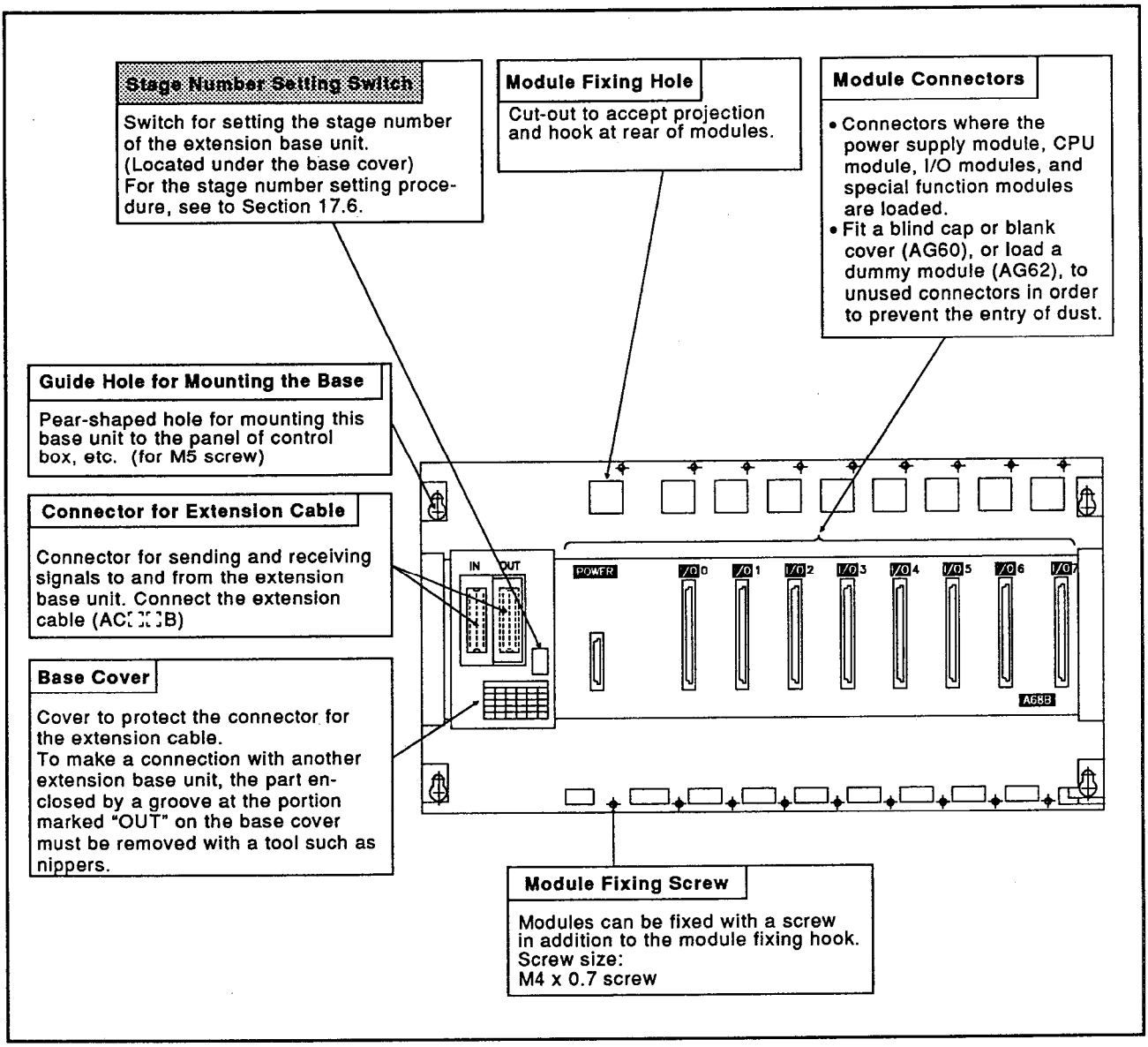
(1) Main base units (A32B, A35B, A38B, A38HB)



POINT

In the case of the A38HB, A38HB is printed beneath the base cover. The A38B has nothing printed here. Use this as a means of distinguishing between base units when mounting modules.

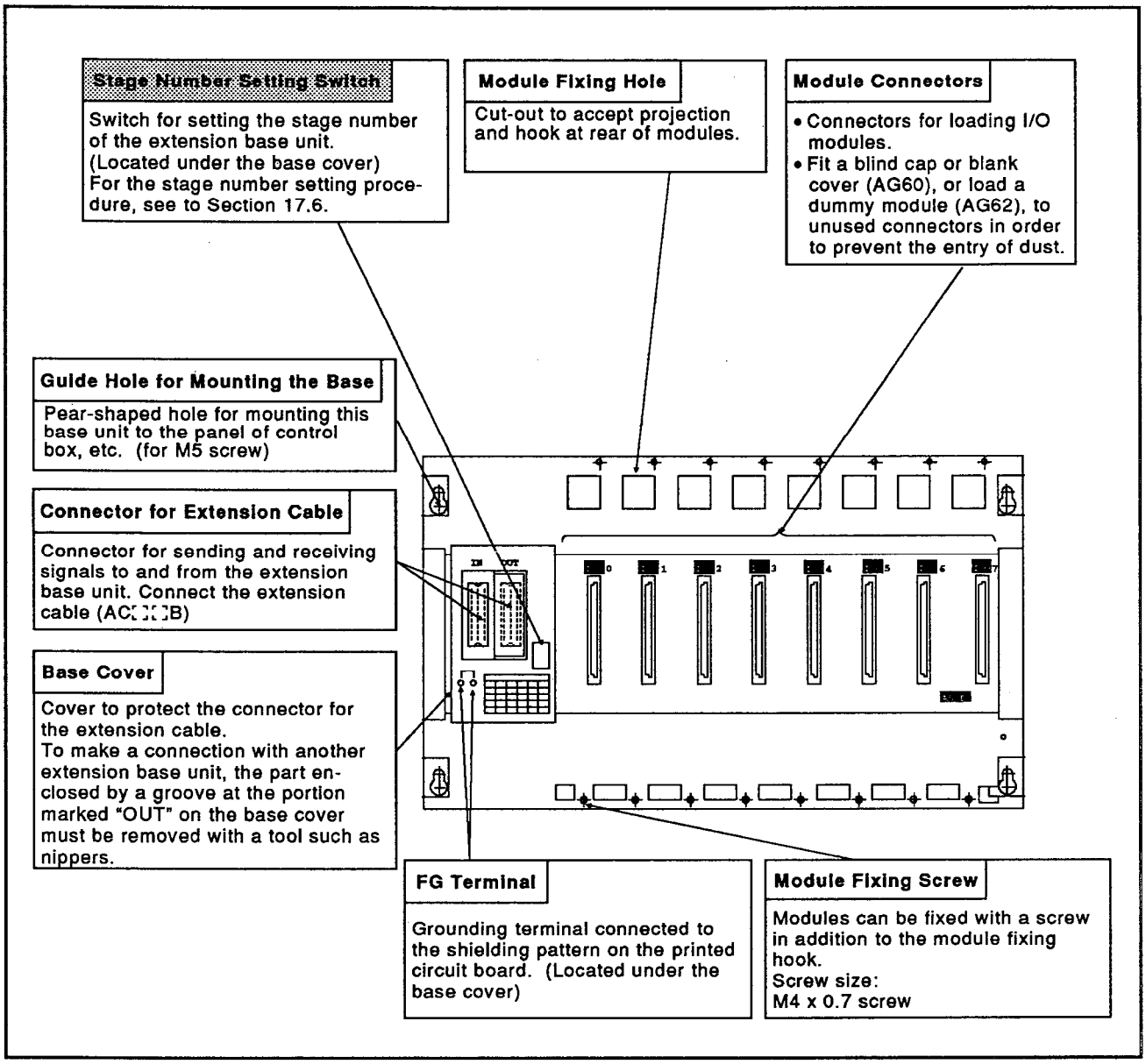
(2) Extension base units (A62B, A65B, A68B)



REMARK

The item indicated by shading ▒▒ must be set before installing the base unit and starting operation.

(3) Extension base units (A52B, A55B, A58B)

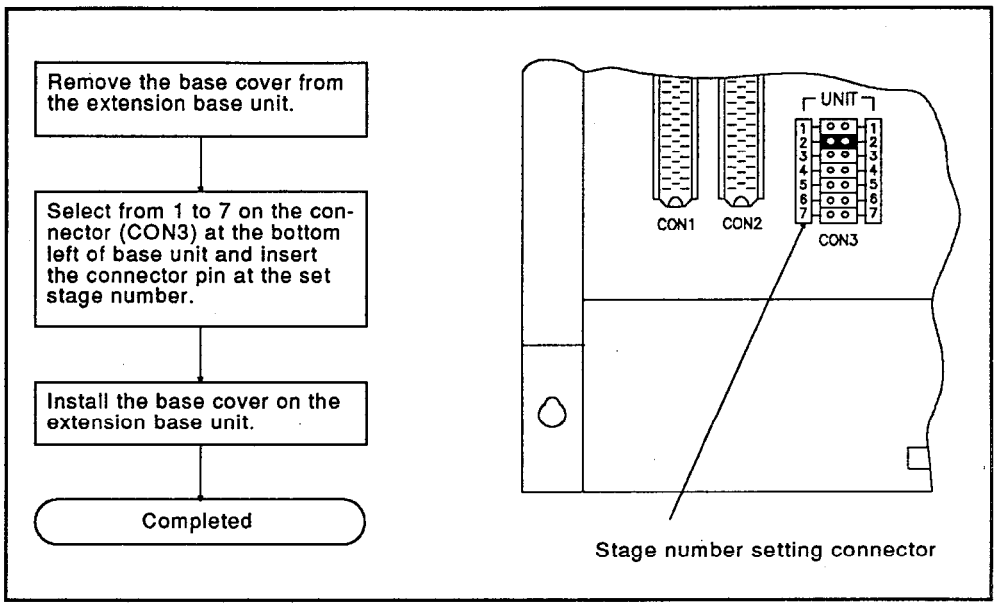


REMARK

The item indicated by shading ▒▒ must be set before installing the base unit and starting operation.

17.6 Setting of Extension Stage Numbers

This section explains the stage number setting procedure for each extension base unit when it is used.



Extension Base Unit Stage Number Setting

	Extension Stage Number Setting						
	1st Stage	2nd Stage	3rd Stage	4th Stage	5th Stage	6th Stage	7th Stage
Setting of the stage number setting connector	UNIT CON3	UNIT CON3	UNIT CON3	UNIT CON3	UNIT CON3	UNIT CON3	UNIT CON3

POINT

Set the stage number setting connector (CON3) to the number, from 1 to 7, which matches the number of extension stages. If the same number has been set for two or more extension base units, or no stage number has been set, erroneous inputs/outputs will result.

18. MEMORY CARDS AND BATTERIES

This section describes the specifications and handling of the memory cards and batteries that can be used with the QnACPU.

18.1 Memory Card Specifications

The specifications of the memory cards that can be used with QnACPU conform to JEIDA Ver. 4.0

(1) SRAM type memory cards

Item \ Model Name	Q1MEM-64S	Q1MEM-128S	Q1MEM-256S	Q1MEM-512S	Q1MEM-1MS	Q1MEM-2MS
SRAM memory capacity before formatting	64 kbytes	128 kbytes	256 kbytes	512 kbytes	1 Mbytes	2 Mbytes
SRAM memory capacity after formatting	59 kbytes	123 kbytes	250.5 kbytes	506 kbytes	1016.5 kbytes	2036 kbytes
Storable number of files	118	128				256
Insertion/removal life	5000 times					
External dimensions (mm) [in]	85.6 x 54 x 3.3 [3.3 x 2.1 x 0.1]					
Weight (g) [lb]	40 [0.09]					

(2) SRAM + E^2PROM type memory cards

Item \ Model Name		Q1MEM-64SE	Q1MEM-128SE	Q1MEM-256SE	Q1MEM-512SE	Q1MEM-1MSE
Memory capacity before formatting	SRAM	32 kbytes	64 kbytes	128 kbytes	256 kbytes	512 kbytes
	E^2PROM	32 kbytes	64 kbytes	128 kbytes	256 kbytes	512 kbytes
Memory capacity after formatting	SRAM	26.5 kbytes	58.5 kbytes	122.5 kbytes	250 kbytes	505.5 kbytes
	E^2PROM	27 kbytes	59 kbytes	123 kbytes	250.5 kbytes	506 kbytes
Storable number of files	SRAM	57	117	128		
	E^2PROM	58	118	128		
EEPROM writing life		10,000 times				
Insertion/removal life		5000 times				
External dimensions (mm) [in]		85.6 x 54 x 3.3 [3.3 x 2.1 x 0.1]				
Weight (g) [lb]		40 [0.09]				

(3) SRAM + flash memory type memory card

Item \ Model Name		Q1MEM-256SF	Q1MEM-512SF	Q1MEM-1MSF	Q1MEM-2MSF
Memory capacity before formatting	SRAM	128 kbytes	256 kbytes	512 kbytes	1 Mbyte
	Flash memory	128 kbytes	256 kbytes	512 kbytes	1 Mbyte
Memory capacity after formatting	SRAM	122.5 kbytes	250 kbytes	505.5 kbytes	1016 kbytes
	Flash memory	*1	*1	*1	*1
Storable number of files	SRAM	128			
	Flash memory	128			
Flash ROM writing life		100,000 times			
Insertion/removal life		5000 times			
External dimensions (mm) [in]		85.6 x 54 x 3.3 [3.3 x 2.1 x 0.1]			
Weight (g) [lb]		40 [0.09]			

*1 The memory capacity of the flash memory after formatting is determined by the specifications of the IC memory card reader/writer used for formatting.

18.2 Handling Memory Cards

(1) Formatting memory cards

All memory cards used with QnACPU must be formatted.
In the case of SRAM + E^2PROM type memory cards and SRAM + flash memory type memory cards, both the RAM and ROM must be formatted.
If the memory card is installed in the QnACPU with only one formatted, the QnACPU detects an error (ICM.OPE.ERROR).
Since the memory card is unformatted on delivery, it must be formatted using a peripheral device capable of GPP functions before use.
The formatting method is as follows: referring to the Type SW□IVD-GPPQ GPP Function Operating Manual (Online), select the PC memory batch processing from the PC menu in the online mode, then select the Format.

(2) Installing the battery in the memory card

On delivery, memory cards are packaged with a battery to provide memory backup for the RAM memory. Before using the RAM memory of the memory card, this battery must be installed.

POINT

Note that even if a battery is installed in the CPU module, the RAM of the memory card will not be backed up if no battery is installed in the memory card.
Similarly, even if a battery is installed in the memory card, the internal memory of the CPU will not be backed up unless a battery is installed in the CPU module.

(3) Switch settings when using a memory card

When using a memory card, set the memory card in/out switch on the connector to which it is connected to the ON position. If it is set to OFF, it will not be possible to use the memory card.

18.3 Battery Specifications (CPU Module, Memory Card Batteries)

(1) CPU module batteries

Item \ Model Name	A6BAT
Nominal voltage	3.6 VDC
Guaranteed life	5 years
Total power interruption time	See Section 20.3.1
Application	Backing up the internal memory, memory back up function
External dimensions (mm) [in]	ϕ 16 x 30 [0.6 x 1.2]

(2) Memory card batteries

Model Name	BR2325 or equivalent
Nominal voltage	3.0 VDC
Guaranteed life	5 years
Total power interruption time	See Section 20.3.2
Application	Backing up the internal memory, memory back up function

18.4 Handling Cautions

The cautions for handling memory cards and batteries in the period between unpacking them and installing them are listed below.

(1) Memory card

(a) Do not drop or bend memory cards or subject them to strong impacts.

(b) Do not expose the memory card to water.

(c) Do not leave memory cards in locations exposed to direct sunlight or close to heating equipment.

(d) Make sure that no contaminants or dust get into the connector.

(e) Do not leave the memory card in hot and humid locations.

(f) To prevent the generation of static electricity, always enclose the memory card in a vinyl case before transporting or storing it.

(g) Do not touch the terminals of the memory card.

(h) When installing the memory card in the CPU module, connect it securely to the connector.

(2) Battery

(a) Do not short the battery.

(b) Do not disassemble the battery.

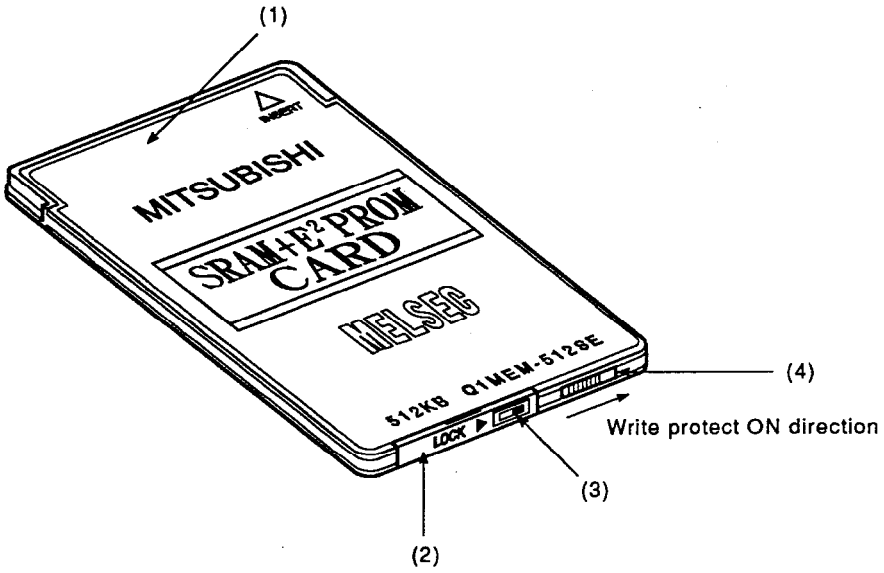
(c) Do not discard the battery into flame.

(d) Do not overheat the battery.

(e) Do not apply solder to the battery poles.

18.5 Memory Card Part Identification

The names of each part of the memory card are indicated below.

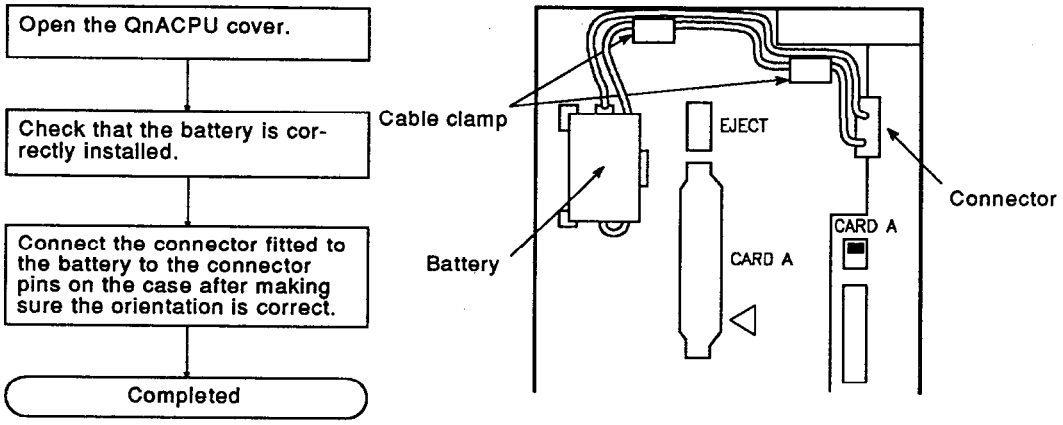


No.	Name	Description	Remark
(1)	Connector	Connector that connects with the CPU module.	
(2)	Battery holder	Holds the lithium battery that backs up the RAM data.	*
(3)	Battery holder locking switch	Switch that locks the battery holder into the memory card. (Locked when at the "LOCK" position.)	
(4)	Write protect switch	Prohibits writing to the memory. Set to OFF on shipping. ON : Data writing prohibited OFF: Data writing enabled	*

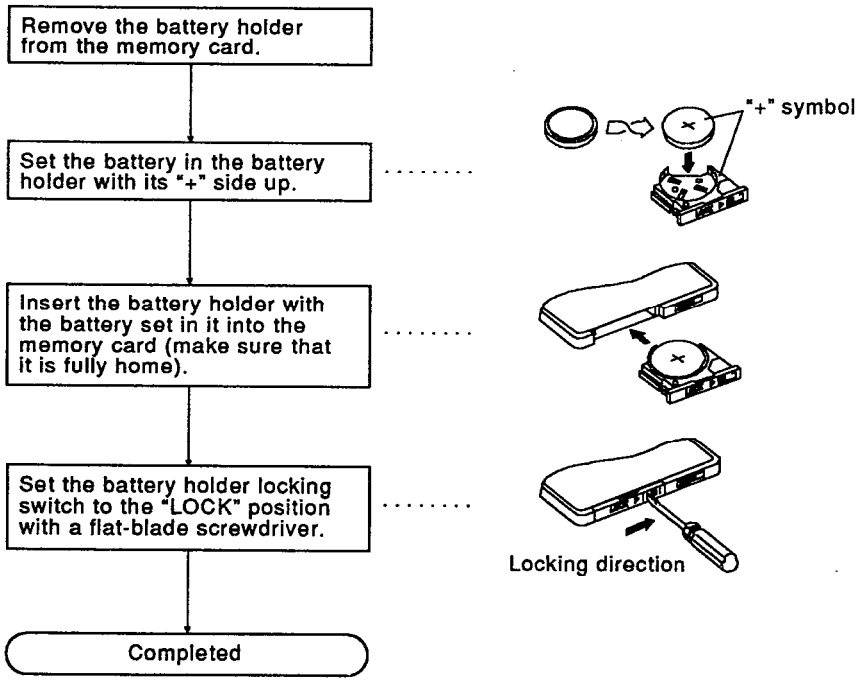
* Must be set before writing a program and before starting operation.

18.6 Installing Batteries (CPU Module, Memory Card Batteries)

(1) Since the battery for the CPU module is shipped with the battery connector disconnected, this connector must be connected according to the procedure indicated below.



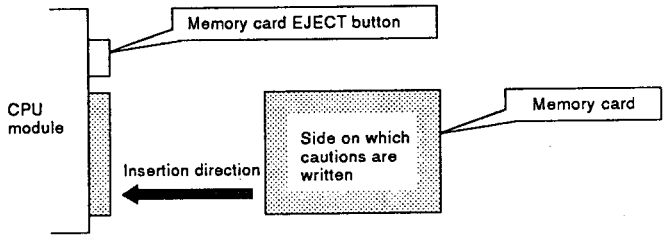
(2) Since the battery for the memory card is shipped disconnected from the battery holder, it must be set in the battery holder by following the procedure described below in order to use the RAM.



18.7 Procedure for Installing/Removing Memory Cards

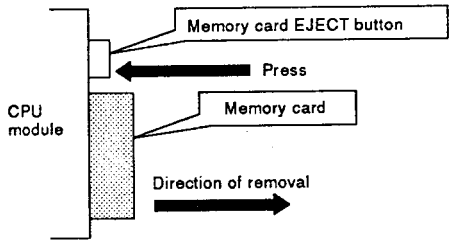
(1) Installing a memory card

To install a memory card in the CPU module while the power is ON, make sure that the orientation of the memory card is correct, then insert it firmly until it projects by the same amount as the EJECT button.
After installing it, set the memory card in/out switch to "ON".
Memory card operation is possible after the LED in the memory card in/out switch has come ON.



(2) Removing a memory card

To remove a memory card from the CPU module while the power is ON, first set the memory card in/out switch to "OFF".
After the LED in the switch has gone OFF, press the memory card EJECT button and remove the memory card.



POINTS

(1) When a memory card is installed, the QnACPU performs mount processing again, and this increases the scan time by a maximum of several tens of seconds.

(2) When the memory card is being used by the system or a program, the LED of the memory card in/out switch may not go OFF even if the switch is set to OFF.

(3) If the memory card is installed or removed after switch ON the memory card in/out switch while the power is ON, the memory contents of the memory card will be destroyed.

(3) Memory card remove/insert prohibit flag (special relays SM605, SM625)

Apart from operating the memory card in/out switch, there is another way to remove or install a memory card, i.e., by turning special relays SM605 (memory card A) and SM625 (memory card B) which are used for the card remove/insert prohibit flag, ON and OFF. When removal/insertion prohibited is set with the remove/install prohibit flag, removal or insertion of a memory card is still prohibited even with the memory card in/out switch set to "ON".
The relationship between the memory card in/out switch and the memory card remove/insert prohibit flag is shown in the table below.

		Memory Card In/Out Switch	
		ON (Removal/Insertion Prohibited)	**OFF (Removal/Insertion Permitted)**
Memory card remove/insert prohibit flag	ON (removal/insertion prohibited)	Removal/insertion prohibited	Removal/insertion prohibited
	OFF (removal/insertion permitted)	Removal/insertion prohibited	Removal/insertion permitted

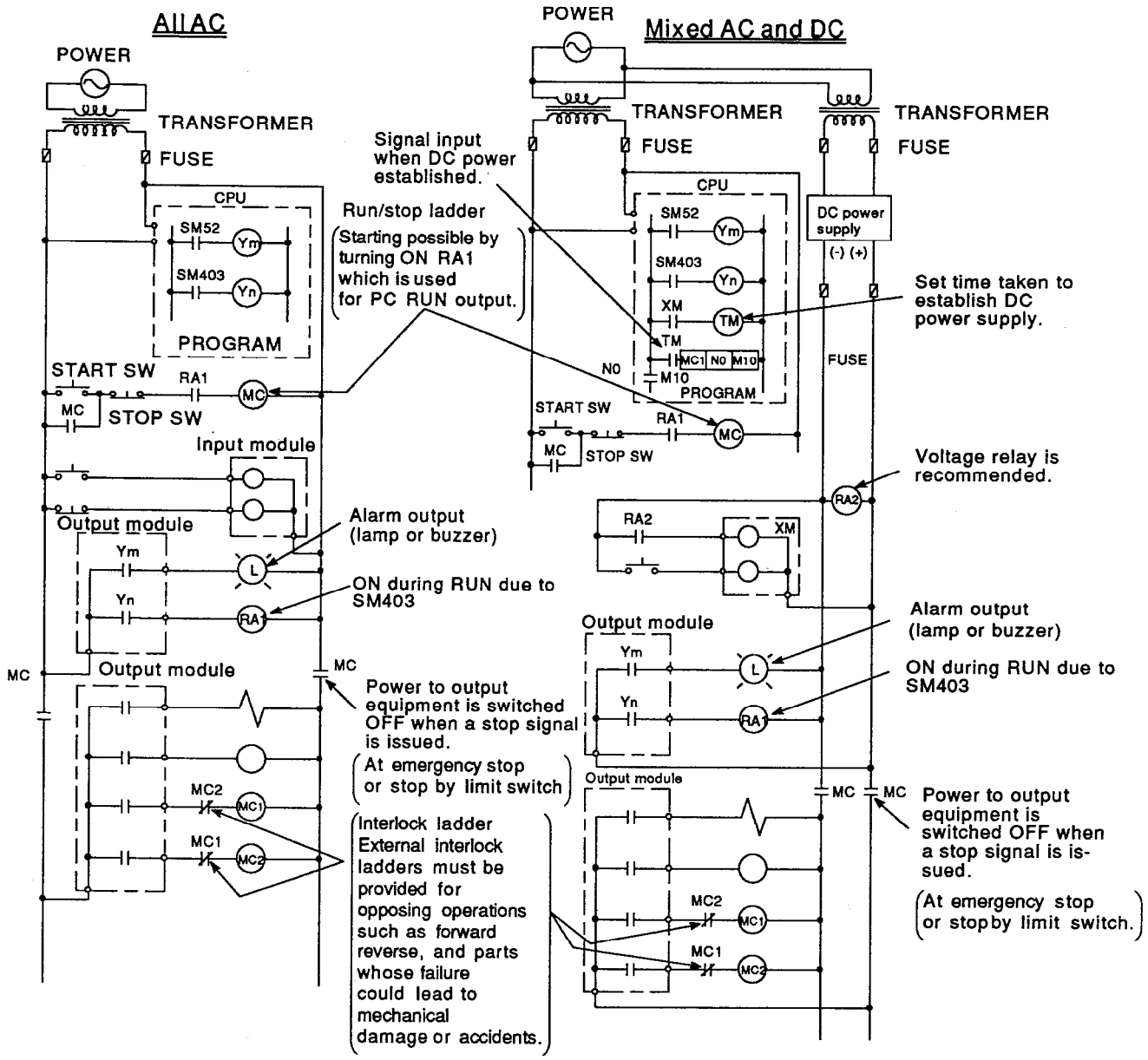
19. LOADING AND INSTALLATION

This chapter describes the loading and installation procedures and precautions to obtain the maximum system reliability and performance.

19.1 Safety Considerations

When the system power supply is turned ON or OFF, process outputs may temporarily become abnormal due to the difference between the delay time and rise time of the power supply of the programmable controller itself and the external power supply (especially DC) for processing.
For example, when the PC power supply is turned ON after turning ON the process external power supply to a DC output module, the DC output module may temporarily generate erroneous outputs when the PC power is turned ON. To avoid this problem, the ladder must be designed so that the PC power supply is turned ON first.
Incorrect operation may also result in the event of an external power supply fault or programmable controller fault.
In order to prevent such faults from causing erroneous operation of the entire system, and also for safety reasons, configure ladders (such as emergency stop ladders, protective ladders, and interlock ladders) that prevent machine damage or accidents in the event of erroneous operation external to the programmable controller.
An example ladder for a system design based on this concept is shown on the following page.

(1) System design ladder example



The power-on procedure is as follows:

All AC

(1) Switch ON power.

(2) Set CPU to RUN.

(3) Switch on the START SW.

(4) When the magnetic contactor (MC) comes ON, output equipment is powered and may be driven from the program.

Mixed AC and DC

(1) Switch ON power.

(2) Set CPU to RUN.

(3) When DC power is established, RA2 turns ON.

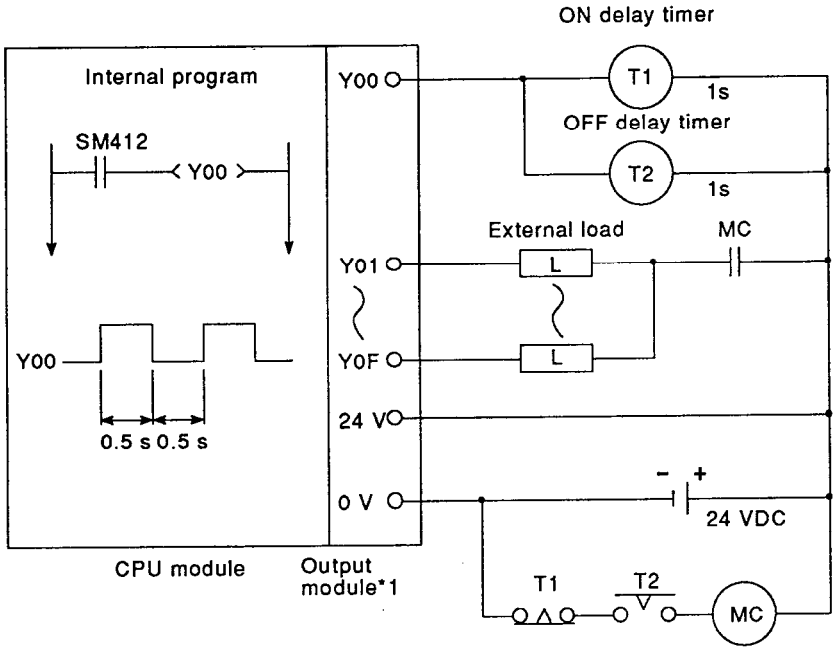
(4) Timer (TM) times out after the DC power reaches 100 %.
(The set value of TM should be the period of time from when RA2 switches ON to the establishment of 100 % DC voltage. Set the set value to 0.5 seconds.)

(5) Switch on the Start SW.

(6) When the magnetic contactor (MC) comes ON, output equipment is powered and may be driven from the program.
(When a voltage relay is used for RA2, the timer (TM) is not necessary in the program.)

(2) Fail-safe measures against PC failures

Problems with the CPU or memory can be detected by the self diagnosis function.
However, problems with I/O control may not be detected by the CPU. If such a problem arises, all I/O points may turn ON or OFF, depending on the nature of the problem, and it may not be possible to maintain normal operating conditions and operating safety.
Although Mitsubishi PCs are manufactured under strict quality control, they could fail or operate abnormally for unspecifiable reasons. To prevent the abnormal operation of the whole system, machine breakdown, and accidents, configure a fail-safe ladder external to the PC.
The following is an example of a fail-safe ladder.



*1 Y00 repeatedly turns ON and OFF at 0.5 second intervals. Use a no-contact output module (transistor in the example shown above).

19.2 Installation Environment

Do not install a QnACPU system at a location subject to any of the following environmental conditions:

(1) Ambient temperatures outside the range of 0 to 55 °C.

(2) Ambient humidity outside the range of 10 to 90 % RH.

(3) Dew condensation due to sudden temperature changes.

(4) Corrosive and/or combustible gasses.

(5) High levels of conductive powders such as dust or iron filings, oil mist, salt, or organic solvents.

(6) Direct sunlight.

(7) Strong electrical or magnetic fields.

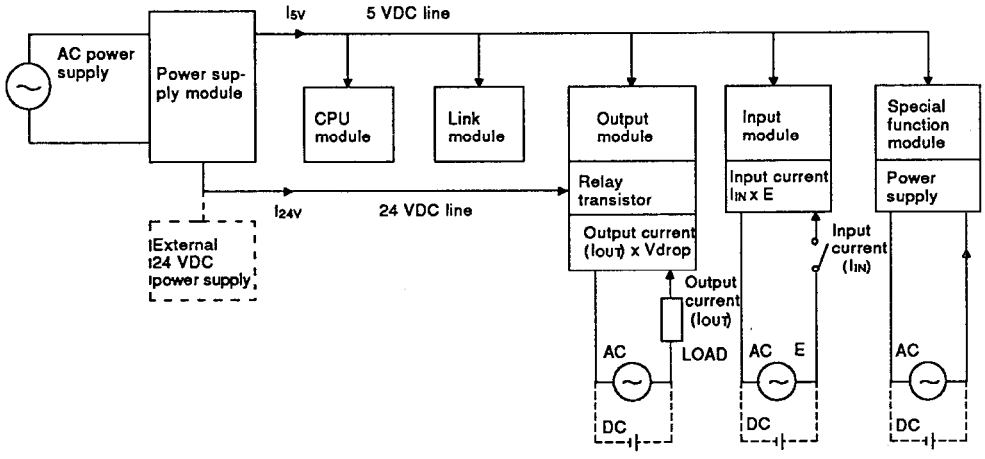
(8) Vibration and shock transmitted directly to the system.

19.3 Calculation of Heat Generated by the Programmable Controller System

The operating ambient temperature of the PC must be kept below 55 °C. In order to plan a heat dissipating design for the panel that houses the equipment, the average power consumption (heat generation) of the devices and equipment housed in the panel must be known. Therefore, the method for determining the average power consumption of a QnACPU system is described here.
Calculate the temperature rise inside the panel from the power consumption.

Average power consumption

Power is consumed by the following PC components:



(1) Power consumption of a power supply module

Approximately 70 % of the power supply module current is converted into power with the remaining 30 % dissipated as heat, i.e., 3/7 of the output power is used.

$$Wpw = \frac{3}{7} \{(I_{5V} \times 5) + (I_{24V} \times 24)\} \quad (W)$$

where, I_{5V} = 5 VDC logic ladder current consumption of each module

I_{24V} = average current consumption of 24 VDC power supply for output module internal consumption (with an average number of points switched ON)

..... Power supply modules for which 24 VDC is supplied from an external source and which have no 24 VDC output are inapplicable.

(2) Total 5 VDC power consumption for each module

The 5 VDC ladder power of the power supply module is the power consumption of each module.

$$W_{5V} = I_{5V} \times 5 \quad (W)$$

(3) Total 24 VDC output module power consumption (with an average number of points switched ON)

The average power of the 24 VDC output ladder of the power supply module is the total power consumption of each module.

$$W_{24V} = I_{24V} \times 24 \quad (W)$$

(4) Average power consumption due to voltage drop at output ladder of output module (with an average number of points switched ON)

W_{OUT} = I_{OUT} x Vdrop x average number of outputs on at one time (W)

where, I_{OUT} = output current (actual operating current) (A)

Vdrop = voltage dropped across each output load (V)

(5) Power consumption of input ladders (with an average number of points switched ON)

W_{IN} = I_{IN} x E x average number of inputs on at one time (W)

where, I_{IN} = input current (effective value for AC) (A)

E = input voltage (actual operating voltage) (V)

(6) Power consumption of the special function module power supply is expressed as:

$$Ws = I_{5V} \times 5 + I_{24V} \times 24 + I_{100V} \times 100 \ (W)$$

The sum of the above values is the power consumption of the entire PC system.

$$W = W_{PW} + W_{5V} + W_{24V} + W_{OUT} + W_{IN} + W_S \ (W)$$

Further calculations are necessary to work out the power dissipated by the other equipment in the panel.

Generally, the temperature rise in the panel is expressed as:

$$T = \frac{W}{UA} \ [°C]$$

where, W = power consumption of the entire PC system (obtained as shown above)

A = panel inside surface area (m^2)

U = (if the panel temperature is controlled by a fan, etc.) ... 6
(if panel air is not circulated) 4

POINT

If the temperature rise inside the panel exceeds the stipulated range, you are recommended to install a heat exchanger in the panel to lower the temperature.
If an ordinary ventilation fan is used, dust will be sucked in along with the air from outside the panel and this may affect the performance of the PC.

19.4 Mounting of Base Unit

This chapter describes the mounting cautions for the main base unit and extension base units.

19.4.1 Mounting cautions

The cautions for mounting the PC to a panel, etc. are presented below:

(1) To improve ventilation and make it easy to replace modules, provide 80 mm (3.15 in) or more of clearance around the PC.

(2) Do not mount the base unit vertically or horizontally since this will obstruct ventilation.

(3) Ensure that the base unit mounting surface is uniform to prevent strain. Application of excessive force to the printed circuit boards will result in incorrect operation. Therefore, mount the base unit on a flat surface.

(4) Avoid mounting the base unit close to vibration sources such as large magnetic contactors and no-fuse breakers. Install the base unit in another panel or distance the base unit from the vibration source.

(5) Provide a wiring duct if necessary.

However, if the dimensions from the top and bottom of the PC are less than those shown in Fig. 19.1, note the following points:

(a) When the duct is located above the PC, the height of the duct should be 50 mm (1.97 in) or less to allow for sufficient ventilation.
Set its distance from the top of the PC so as to allow the hook latches on the tops of modules to be pressed.
If the hook latch on the top of a module cannot be pressed, the module cannot be replaced.

(b) When the duct is located under the PC, install the duct so that optical fiber cables or coaxial cables may be connected and also consider the minimum bending radius of the cable.

(6) If an equipment which generates noise or heat is positioned in front of the PC (i.e., mounted on the back side of a panel door), allow a clearance of 100 mm (3.94 in) or more between the PC and the equipment.
Also allow a clearance of 50 mm (1.97 in) or more between the right/left side of a base unit and this equipment.

19.4.2 Mounting

This section explains how to mount for main and extension base units.

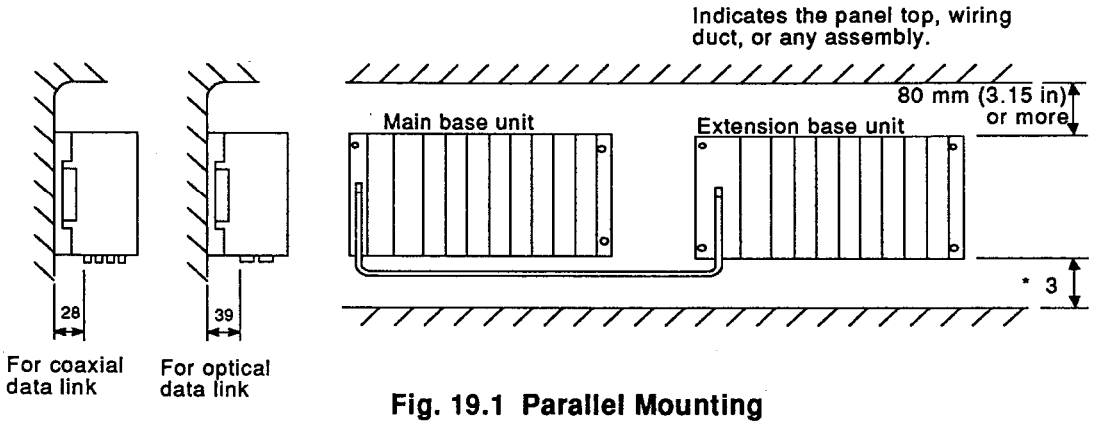


Fig. 19.1 Parallel Mounting

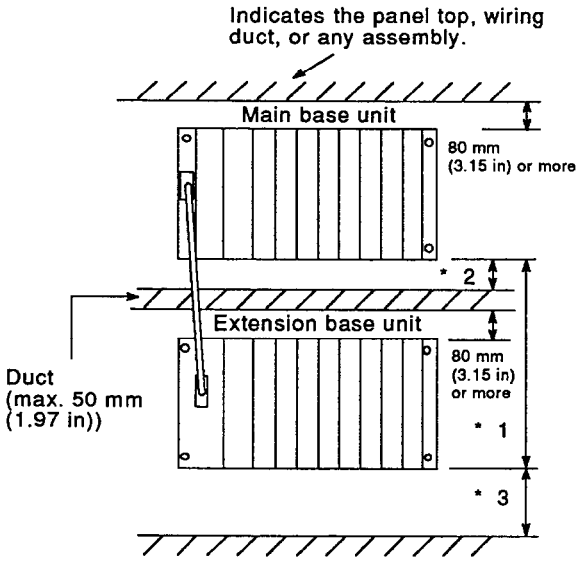


Fig. 19.2 Serial Mounting

*1:.... Depends on the length of the extension cable as indicated below.

For Type AC06B cable ... 450 mm (17.72 in) or less
For Type AC12B cable ... 1050 mm (41.34 in) or less
For Type AC30B cable ... 2850 mm (112.21 in) or less

*2:.... When no link module is used. ..... 50 mm (1.97 in) or more

When ϕ 4.5 mm (0.18 in) dia. optical fiber cable or coaxial cable is used. ..... 100 mm (3.94 in) or more

When ϕ 8.5 mm (0.33 in) dia. optical fiber cable is used. ..... 130 mm (5.19 in) or more

*3:.... When the link unit is not used..... 50 mm (1.97 in) or more

When ϕ 4.5 mm (0.18 in) dia. optical fiber cable or coaxial cables used. ..... 100 mm (3.94 in) or more

When ϕ 8.5 mm (0.33 in) dia. optical fiber cable is used. ..... 130 mm (5.19 in) or more

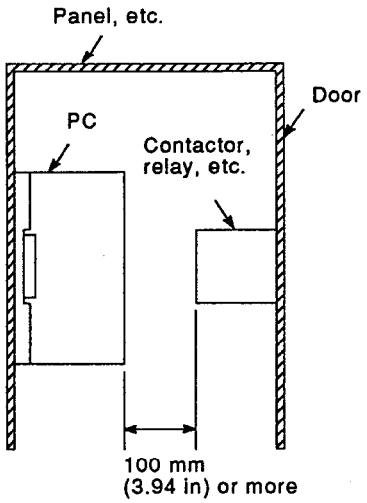


Fig. 19.3 Minimum Front Clearance with Panel Door Shut

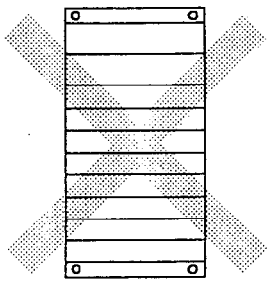


Fig. 19.4 Vertical Mounting (Not allowed)

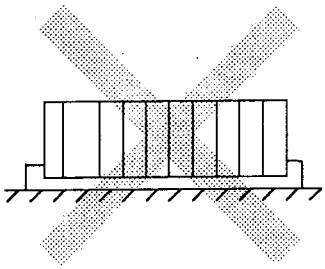


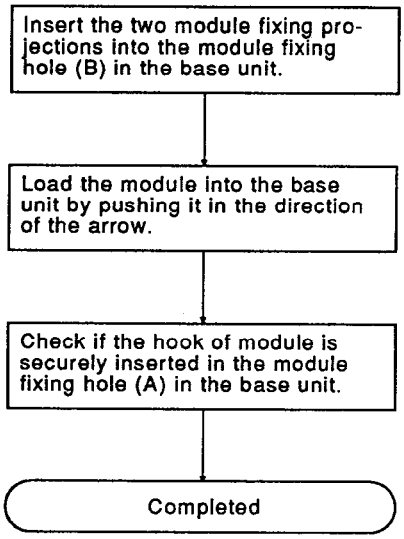
Fig. 19.5 Horizontal Mounting (Not allowed)

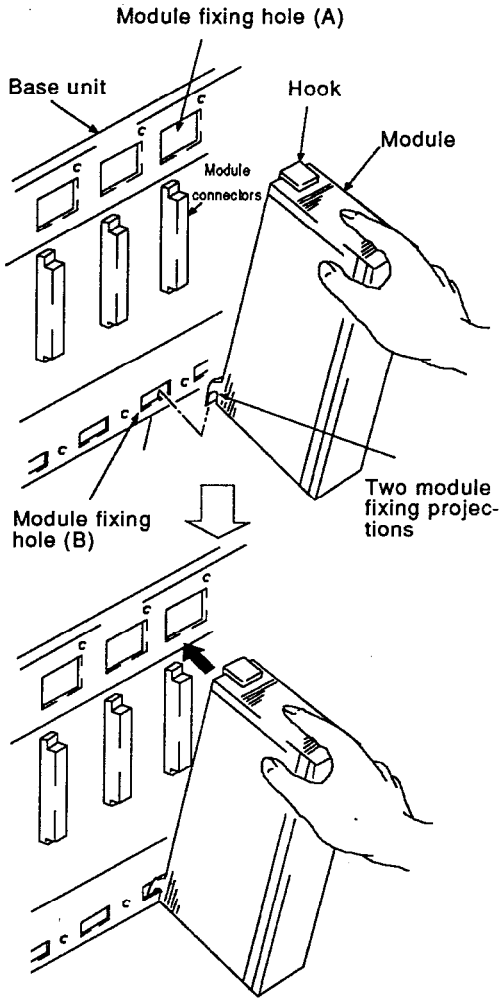
19.5 Mounting and Removing Modules

This section explains procedures for mounting and removing power supply modules, CPU modules, I/O modules, special function modules, etc. to and from the base unit.

(1) Mounting a module

The procedure for mounting a module is described below.

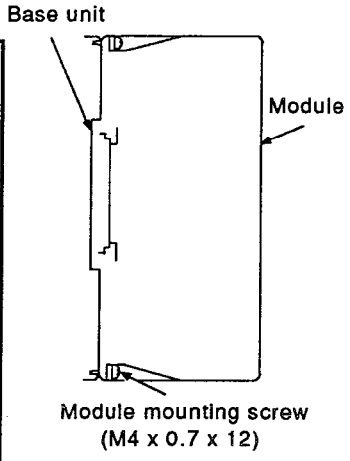




POINTS

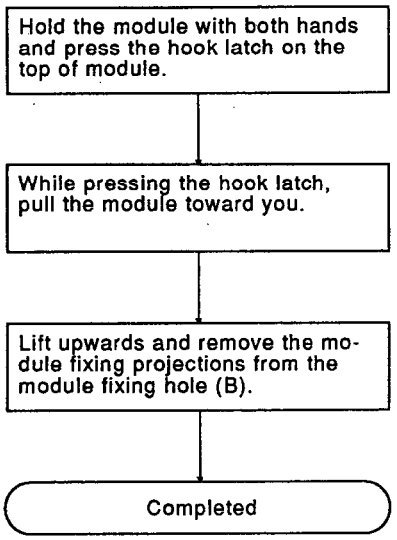
(1) To secure the module, be sure to insert the module fixing projections into the module fixing hole (B). If the module is forcibly secured without inserting the projections, the pins in the module connector may be bent or damaged.

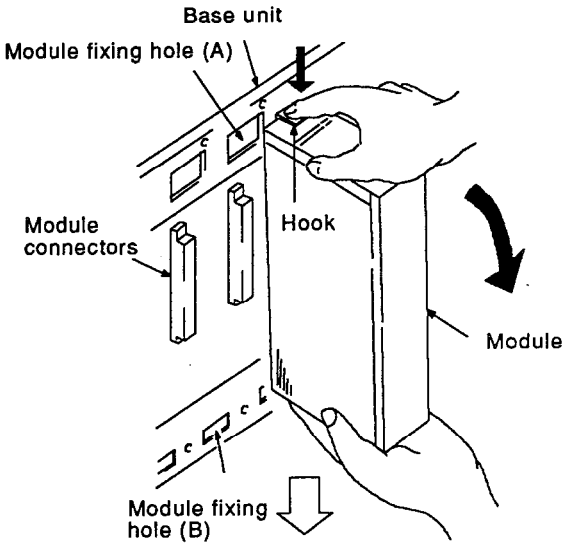
(2) When the base unit is used at a location subject to especially large vibrations and/or shock, screw the module to the base. The applicable screw size is M4(0.16) x 0.7(0.03) x x 12 mm (0.47 in). See the figure at right.

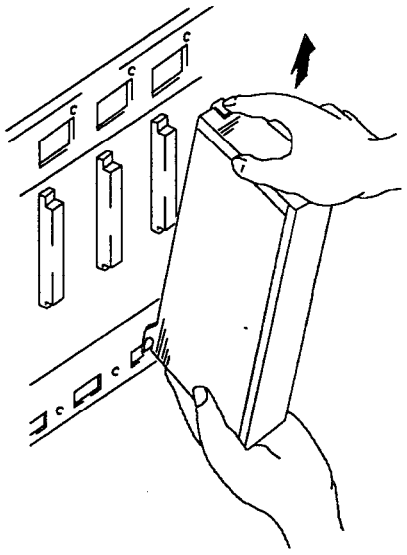


(2) Removing a module

The procedure for removing a module is described below.





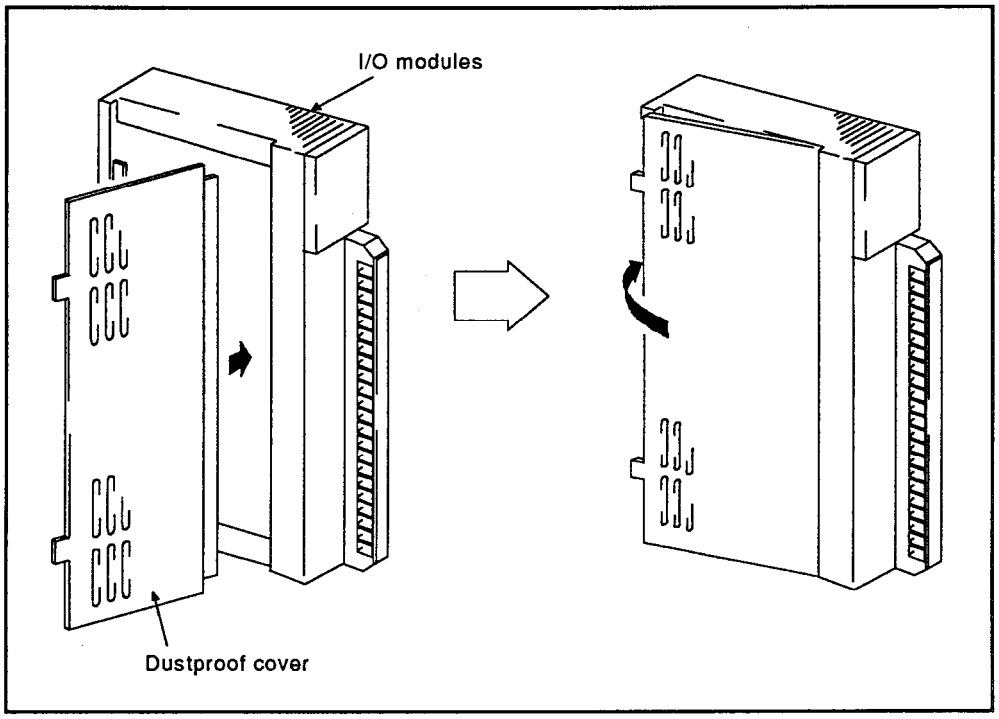


POINT

To remove the module, be sure to disengage the hook from the module fixing hole (A) and then remove the module fixing projections from the module fixing hole (B). If the module is forcibly removed, the hook or module fixing projections will be damaged.

19.6 Fitting the Dustproof Cover

When using an A52B, A55B or A58B a dustproof cover (supplied with the base unit) must be fitted to the I/O module loaded at the left end in order to prevent foreign matter from entering the I/O module. If the dustproof cover is not fitted, foreign matter will enter the I/O module, causing it to fail. The method for fitting the dustproof cover is explained below.



To insert the dustproof cover into the I/O module, first insert the cover at the terminal side and then press it against the I/O module as shown in the figure.

19.7 Wiring

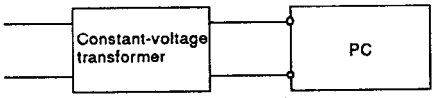
This section describes the wiring cautions for the system.

19.7.1 Wiring cautions

Cautions for wiring the power cable and I/O cables.

(1) Wiring of the power supply

(a) When voltage fluctuations are larger than the specified value, connect a constant-voltage transformer.



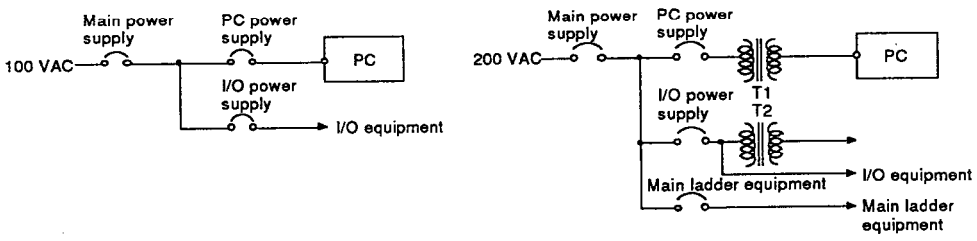
(b) Use a power supply which generates minimal noise between wires and between the PC and ground.
If excessive noise is generated, connect an insulating transformer.

(c) When a power transformer or insulating transformer is employed to reduce the voltage from 200 VAC to 100 VAC, use one with a capacity greater than those indicated in the following table.

Power Supply Module Model Name	Transformer Capacity
A61P, A61PEU	110VA x n
A62P, A62PEU	110VA x n
A65P	110VA x n
A66P	95VA x n

n: stands for the number of power supply modules.

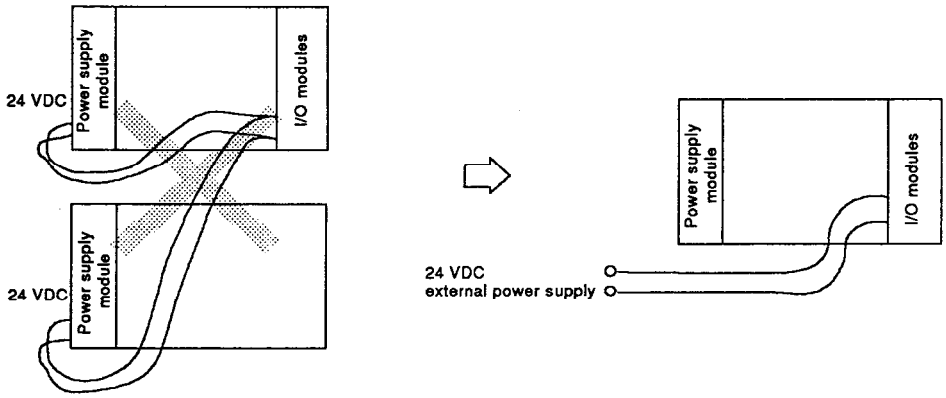
(d) When wiring, separate the PC power supply from the I/O and power equipment as shown below.



REMARK

As a safety measure, install a switch for use with "online I/O module change" only to each of the corresponding modules and equipment.

(e) Note on using the 24 VDC output of the A62P(EU), A65P and A66P power supply module.
To protect the power supply modules, do not supply one I/O module with 24 VDC from several power supply modules connected in parallel. From several power supply modules connected in parallel. If the 24 VDC output capacity is insufficient with one power supply module, supply 24 VDC from the external 24 VDC power supply as shown below:

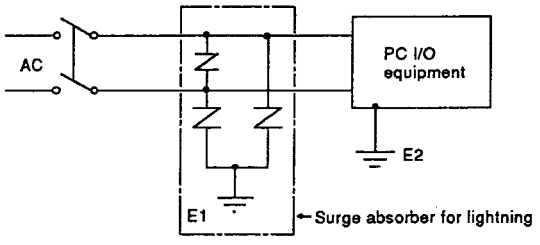


(f) Twist the 100 VAC, 200 VAC, and 24 VDC cables as closely as possible. Connect modules with the shortest possible wire lengths.

(g) To minimize voltage drop, use the thickest (max. 2 mm^2 (14 AWG)) wires possible for the 100 VAC, 200 VAC, and 24 VDC cables.

(h) Do not bundle the 100 VAC and 24 VDC cables with main-ladder wires or the I/O signal wires (high-voltage, large-current), or lay these cables and wires close to each other when wiring. If possible, separate the cables and wires by a distance of more than 100 mm (3.94 in).

(i) As a lightning-protection measure, connect a surge absorber as shown below.



POINTS

(1) Ground the surge absorber (E1) and the PC (E2) separately from each other.

(2) When selecting a surge absorber, make sure that the maximum permitted circuit voltage for the surge absorber will not be exceeded.

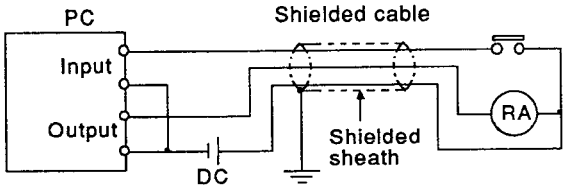
(2) Wiring of I/O equipment

(a) The applicable size of wire for connection to the terminal block connector is 0.75 to 2 mm^2. However, it is recommended to use wires of 0.75 mm^2 for convenience.

(b) Separate the input and output lines.

(c) I/O signal wires must be at least 100 mm (3.94 in) away from high-voltage and large-current main ladder wires.

(d) If the I/O signal wires cannot be separated from the main ladder wires and power wires, ground at the PC side with shielded cables. Under some conditions, it may be preferable to ground at the other side.



(e) If wiring has been done with piping, ground the piping.

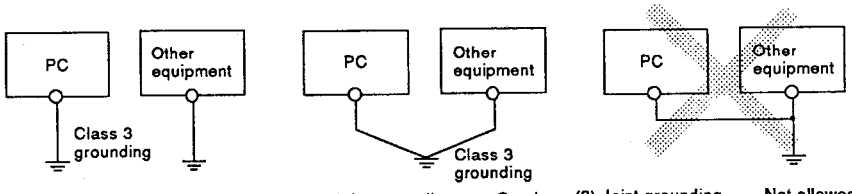
(f) Separate the 24 VDC I/O cables from the 100 VAC and 200 VAC cables.

(g) If wiring over 200 mm (7.87 in) or longer distances, problems can be caused by leakage currents due to line capacity. Take corrective action as described in Section 21.4.

(3) Grounding

(a) Ground the PC independently if possible. Class 3 grounding should be used (grounding resistance 100 Ω or less).

(b) If independent grounding is impossible, use the joint grounding method as shown in the figure below (2).



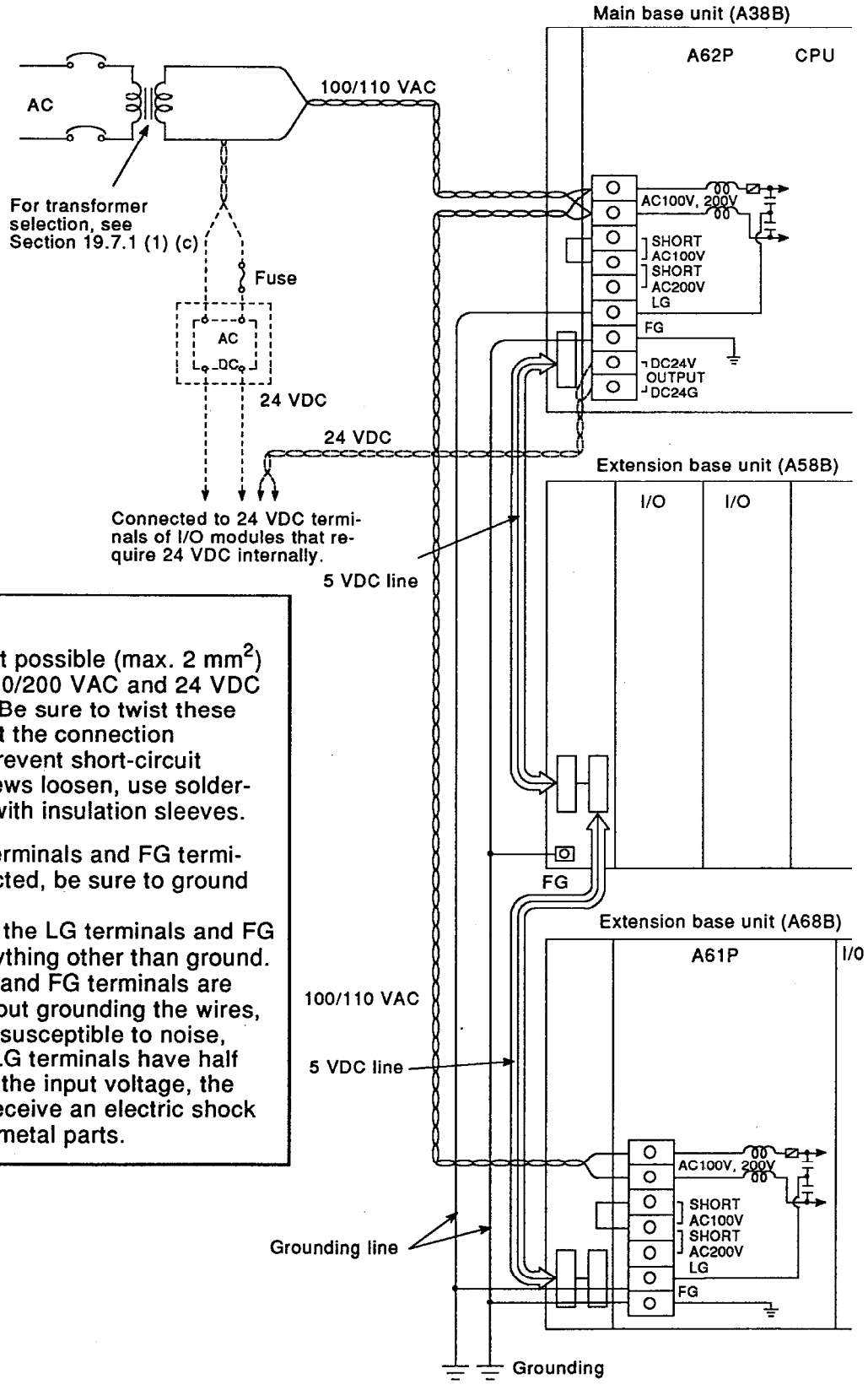
(1) Independent grounding .... Best (2) Joint grounding .... Good (3) Joint grounding .... Not allowed

(c) Use a wire with a cross-sectional area of at least 2 mm^2 for grounding.
Make the grounding point as close to the PC as possible so that the grounding wire is not too long.

(d) If any malfunction occurs due to grounding, disconnect either or both of the LG and FG terminals of the base unit from the ground.

19.7.2 Wiring to base units

This section explains the wiring of power lines and grounding lines to the main and extension base units.



POINTS

(1) Use the thickest possible (max. 2 mm^2) wires for the 100/200 VAC and 24 VDC power cables. Be sure to twist these wires starting at the connection terminals. To prevent short-circuit should any screws loosen, use solderless terminals with insulation sleeves.

(2) When the LG terminals and FG terminals are connected, be sure to ground the wires.
Do not connect the LG terminals and FG terminals to anything other than ground. If LG terminals and FG terminals are connected without grounding the wires, the PC may be susceptible to noise, also since the LG terminals have half the potential of the input voltage, the operator may receive an electric shock when touching metal parts.

20. MAINTENANCE AND INSPECTION

This chapter describes the items to be checked in daily and periodic maintenance and inspection in order to maintain the programmable controller in the normal and optimum condition.

20.1 Daily Inspection

Table 20.1 shows the inspection and items which are to be checked daily.

Table 20.1 Daily Inspection

Item	Check Item		Check Point	Criterion of Judgment	Corrective Action
1	Base unit mounting conditions		Check for loose mounting screws and cover.	The base unit should be securely mounted.	Retighten screws.
2	Mounting conditions of I/O module, etc.		Check if the module is disengaged and if the hook is securely engaged.	The hook should be securely engaged and the module should be positively mounted.	Securely engage the hook.
3	Connecting conditions		Check for loose terminal screws.	Screws should not be loose.	Retighten terminal screws.
			Check distance between solderless terminals.	The proper clearance should be provided between solderless terminals.	Correct.
			Check connectors of extension cable.	Connections should not be loose.	Retighten connector mounting screws.
4	CPU module indicator lamps	"POWER" LED	Check that the LED is ON.	ON (OFF indicates an error)	See Section 21.2.2.
		CPU "RUN" LED	Check that the LED is ON during RUN.	ON (OFF indicates an error)	See Section 21.2.3 and 21.2.4.
		CPU "ERROR" LED	Check that the LED is OFF.	OFF (ON or flashing indicates an error)	See Section 21.2.5 and 21.2.6.
		CPU "BAT. ARM" LED	Check that the LED is OFF.	OFF (ON indicates an error)	See Section 21.2.8
		Input LED	Check that the LED turns ON and OFF.	ON when input is ON. OFF when input is OFF. (Display other than above, indicates an error.)	See Section 21.2.9
		Output LED	Check that the LED turns ON and OFF.	ON when output is ON. OFF when output is OFF. (Display other than above, indicates an error.)	See Section 21.2.9

REMARK

If an I/O module has to be replaced when the system is running, use the "online I/O module replacement" function.

20.2 Periodic Inspection

This section explains the inspection items which are to be checked every six months to one year. This inspection should also be performed when the equipment is moved or modified or the wiring is changed.

Table 20.2 Periodic Inspection

Item	Check Item		Checking Method	Criterion of Judgment	Corrective Action
1	Ambient environment	Ambient temperature	Measure with thermometer and hygrometer. Measure corrosive gas.	0 to 55 °C	When the PC is used inside a panel, the temperature inside the panel is the ambient temperature.
		Ambient humidity		10 to 90 %RH	
		Atmosphere		There should be no corrosive gases.	
2	Power supply voltage check		100/200 VAC, 24 VDC, 110 VDC Measure voltage across 100/200 VAC terminals.	85 to 132 VAC 170 to 264 VAC 15.6 to 31.2 VDC 85 to 140 VDC	Change supply power.
3	Fuse		Check if the fuse is blown.	(Preventive maintenance)	Even if a fuse has not blown, the element may have deteriorated due to inrush current, and the fuse should therefore be changed at regular intervals.
4	Mounting conditions	Looseness, play	Move the modules.	The modules should be mounted securely and positively.	Retighten screws. For CPU, I/O, and power supply modules check all connections.
		Ingress of dust or foreign material	Visual check.	There should be no dust or foreign material in the vicinity of the PC.	Remove and clean.
5	Connecting conditions	Check for loose terminal screws	Retighten connector mounting screws.	Screws should not be loose.	Retighten.
		Distance between solderless terminals	Visual check.	The proper clearance should be provided between solderless terminals.	Correct.
		Loose connector	Visual check.	Screws should not be loose.	Retighten screws.
6	Battery		Check that SM51 and SM52 are OFF at a peripheral device in the monitor mode.	(Preventive maintenance)	If no reduction of battery capacity is indicated, change the battery when specified service life is exceeded.

20.3 Replacement of Batteries

Special relays SM51 and SM52 turn ON when the voltage of the battery for program and memory backup falls. Even if these special relays turn ON, neither program data nor the memory backup function are lost immediately. However, if the ON state is overlooked, the PC data contents may be lost.
Replace the battery before the total backup time after special relay SM51 comes ON reaches the stipulated maximum.

POINT

SM51 is the battery voltage drop alarm; data is retained for a stipulated time after it comes ON, but the battery should be replaced as soon as possible.
SM52 indicates an error where the battery is completely discharged; when it comes ON the battery must be changed promptly.

SM51 and SM52 will come ON if there is a voltage drop in the battery of any of the following: the internal memory, memory card A, or memory card B.
In order to determine which of these memories' battery has sustained the voltage drop, check the contents of SD51 and SD52.
When the voltage of any memory's battery drops, the bit in SD51 and SD52 that corresponds to the memory in question turns ON.

SD51, SD52 bit No.	Corresponding Memory
Bit 0	Internal memory
Bit 1, 2	Memory card A
Bit 3, 4	Memory card B

POINT

The relationship between the statuses of the batteries installed in the CPU module and memory cards and memory back up is explained below.
The following two points apply:

(1) The battery installed in the CPU module does not back up the memories of the memory cards.

(2) The batteries installed in the memory cards do not back up the memory of the CPU module.

CPU Module AC Power Supply	CPU Module's Battery	Memory Card Battery	CPU Module Memory	Memory Card Memory
ON	ON	ON	o	o
		OFF	o	o
	OFF	ON	o	o
		OFF	o	o
OFF	ON	ON	o	o
		OFF	o	x
	OFF	ON	x	o
		OFF	x	x

o: Back up possible
x: Back up not possible

The following sections give the battery service life and the battery changing procedure.

20.3.1 Service life of batteries

(1) Service life of CPU module's battery
The CPU module battery life differs depending on the CPU model. The life for each CPU model is shown in Table 20.3.

Table 20.3 CPU Module Battery Life

Battery Life / CPU Model	Battery Life (Total Power Interruption Time) [Hr]		
	Guaranteed Value (MIN)	Actual Service (TYP)	After SM51 is Turned ON
Q2ACPU	1800	14500	48
Q2ACPU-S1	1150	10700	27
Q3ACPU	4000	18000	113
Q4ACPU	1750	6200	44

* The actual service value indicates a typical life time and the guaranteed value indicates the minimum life time.

(2) Service life of memory card batteries
The memory card battery life differs depending on the memory capacity. The life for each memory is shown in Table 20.4.

Table 20.4 Battery Lives of Memory Cards

Battery Life / Memory Card Model	Battery Life (Total Power Interruption Time) [Hr]		
	Guaranteed Value (MIN)	Actual Service (TYP)	After SM51 is Turned ON
Q1MEM-64S	5256	23652	8
Q1MEM-128S	2628	12264	6
Q1MEM-256S	5256	23652	8
Q1MEM-512S	2628	12264	6
Q1MEM-1MS	7008	23652	6
Q1MEM-2MS	2628	12264	6
Q1MEM-64SE	5256	23652	8
Q1MEM-128SE	5256	23652	8
Q1MEM-256SE	5256	23652	8
Q1MEM-512SE	5256	23652	8
Q1MEM-1MSE	2628	12264	6
Q1MEM-256SF	2628	12264	6
Q1MEM-512SF	5256	23652	8
Q1MEM-1MSF	2628	12264	6
Q1MEM-2MSF	7008	19272	6

* The actual service value indicates a typical life time and the guaranteed value indicates the minimum life time.

The guide for preventive maintenance is as follows.

(1) Even if the total power interruption time is less than the guaranteed value in the above table, change the battery after four to five years.
(2) When the total power interruption time has exceeded the guaranteed value in the table above and SM52 has turned ON, change the battery.

20.3.2 Battery replacement procedure

(1) Replacing the CPU module's battery
When the service life of the CPU module's battery has expired, change the battery using the following procedure. Even if the battery is removed, the memory is backed by a capacitor for some time. However, if the time taken to replace the battery exceeds the guaranteed value shown in Table 20.5 below, the contents of the memory may be lost. Therefore, change the battery as quickly as possible. It is also possible to replace the battery while the power supply to the CPU module is ON. In this case the memory contents are backed up by the power supply voltage from the power supply module.

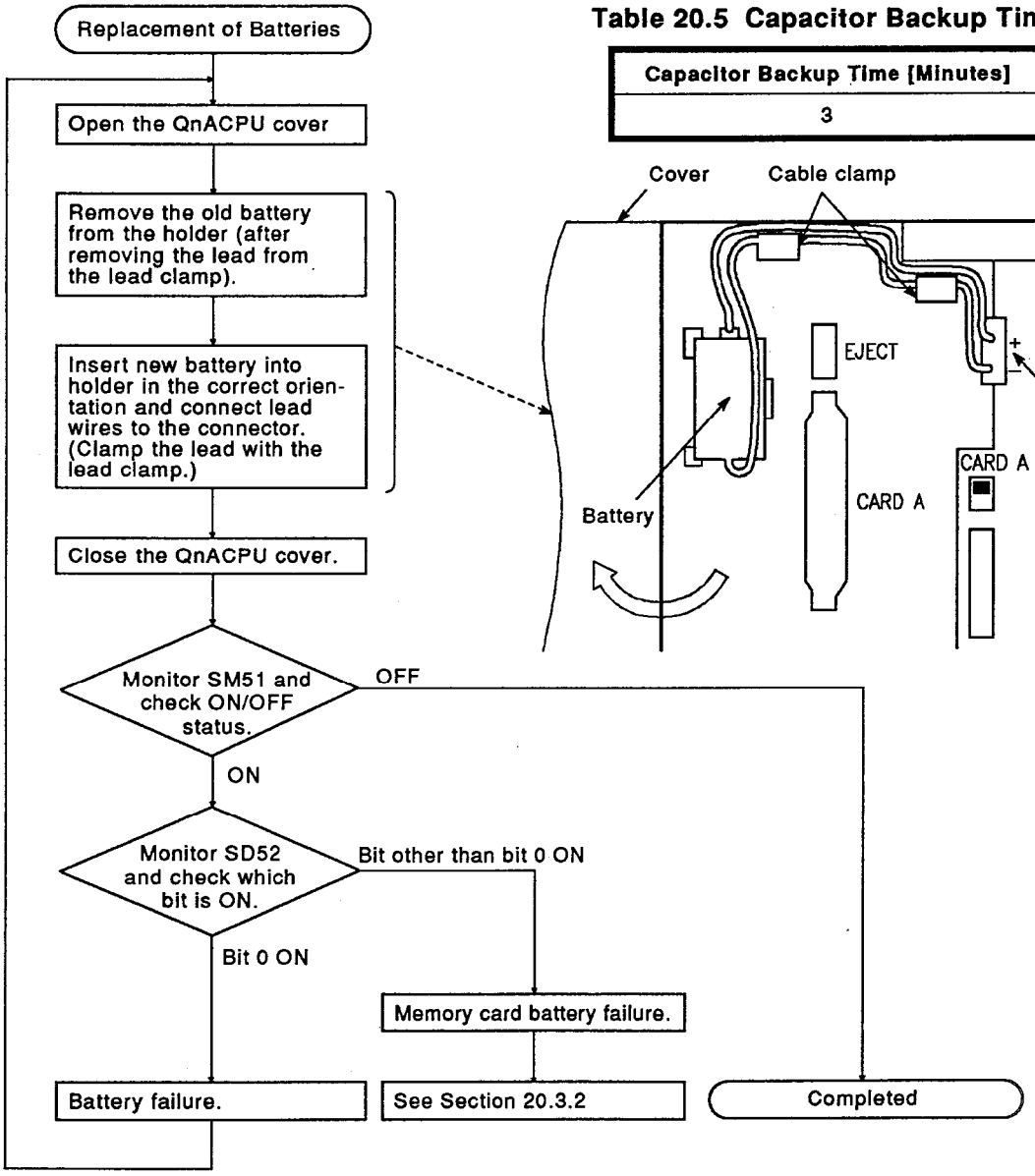


Table 20.5 Capacitor Backup Time

Capacitor Backup Time [Minutes]
3

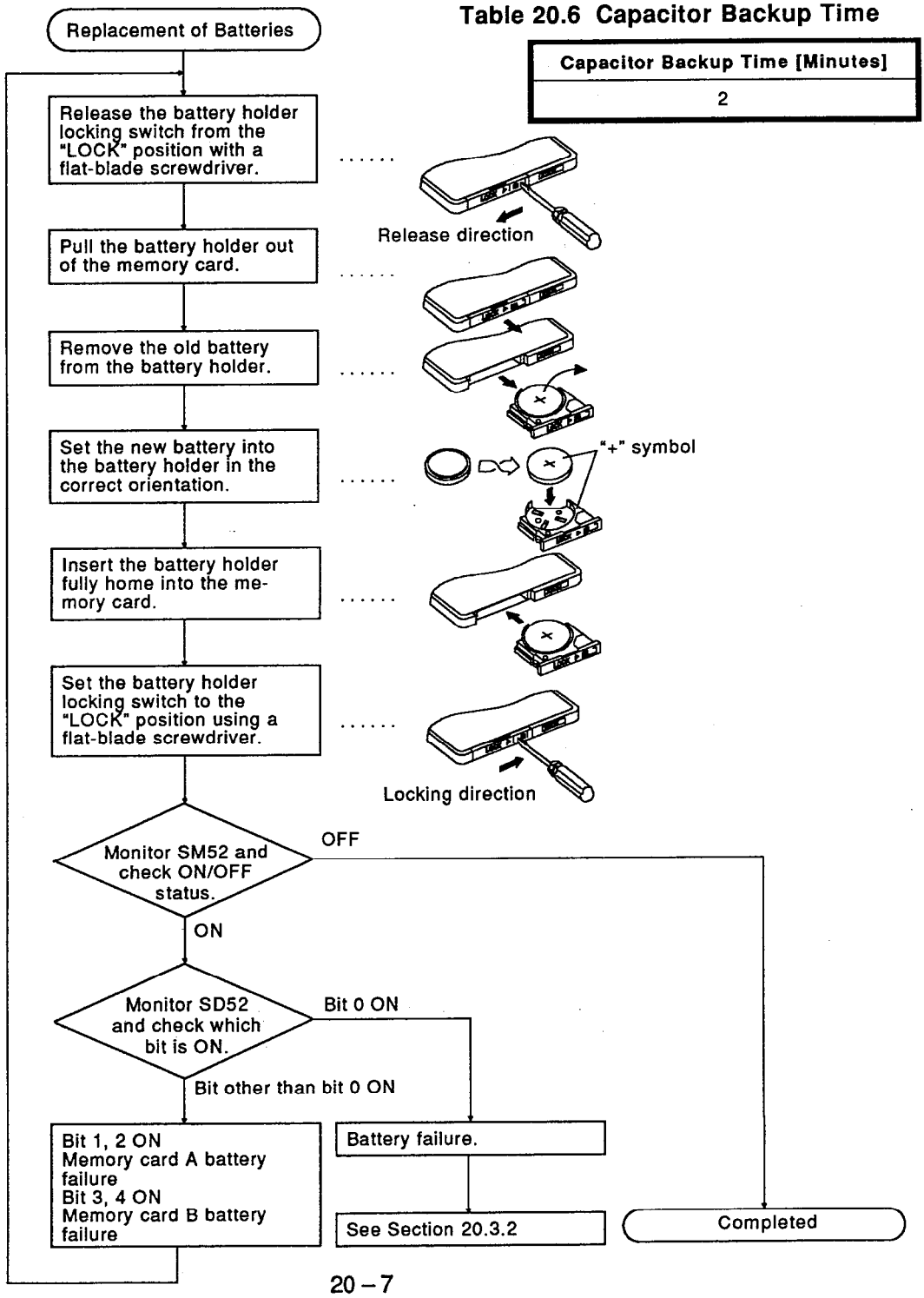
(2) Replacing memory card batteries

When the service life of the battery of a memory card has expired, replace the battery by using the following procedure. Even if the battery is removed, the memory card memory is backed up by a capacitor so that the battery can be replaced while the memory card is out of the CPU module.

However, if the time taken to replace the battery exceeds the guaranteed value indicated in Table 20.6 below, the contents of the memory may be lost. Therefore, change the battery as quickly as possible. It is also possible to replace the battery while the power supply to the CPU module is ON and the memory card is still installed in the CPU module. In this case the memory contents are backed up by the power supply voltage from the power supply module.

Table 20.6 Capacitor Backup Time

Capacitor Backup Time [Minutes]
2

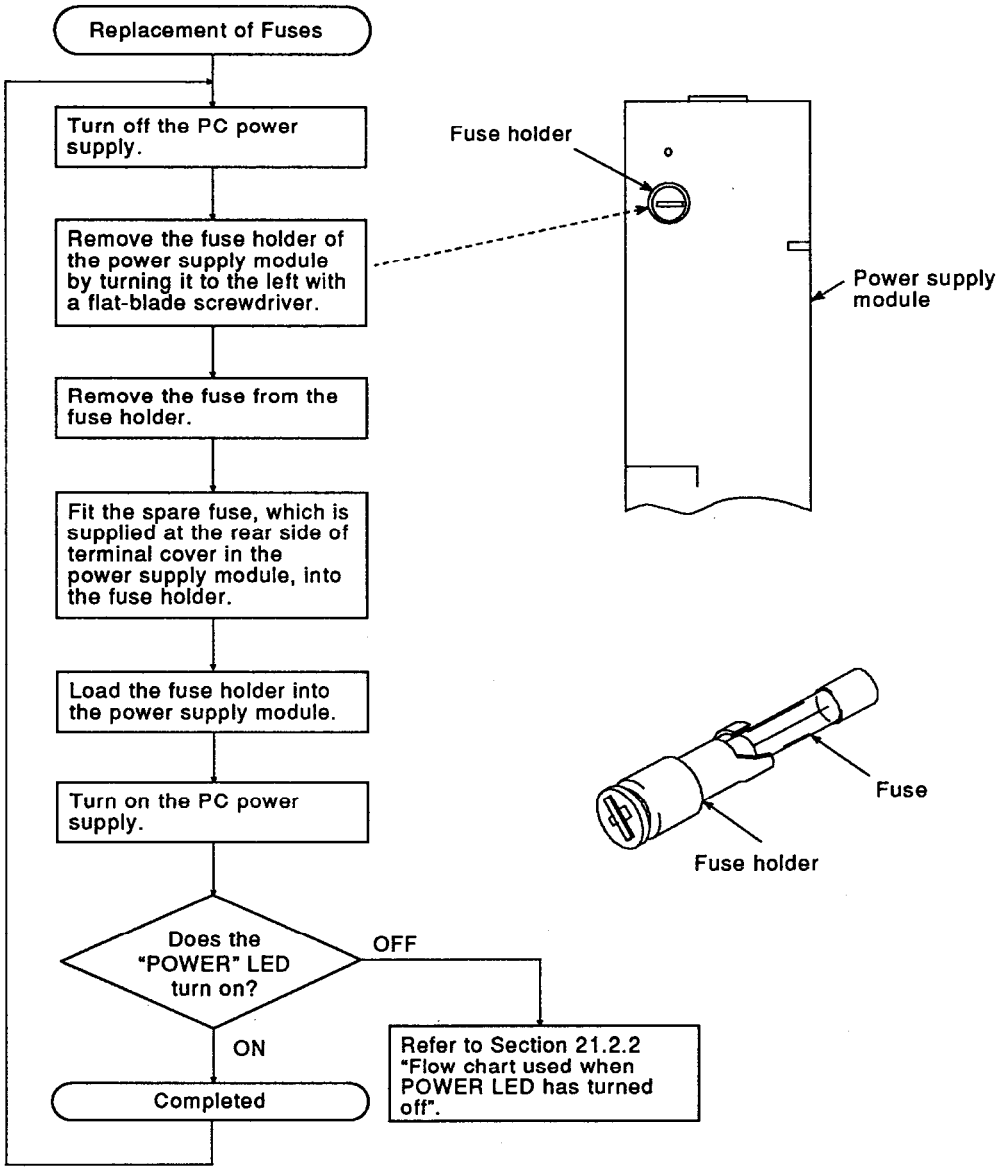


20.4 Replacement of Fuses

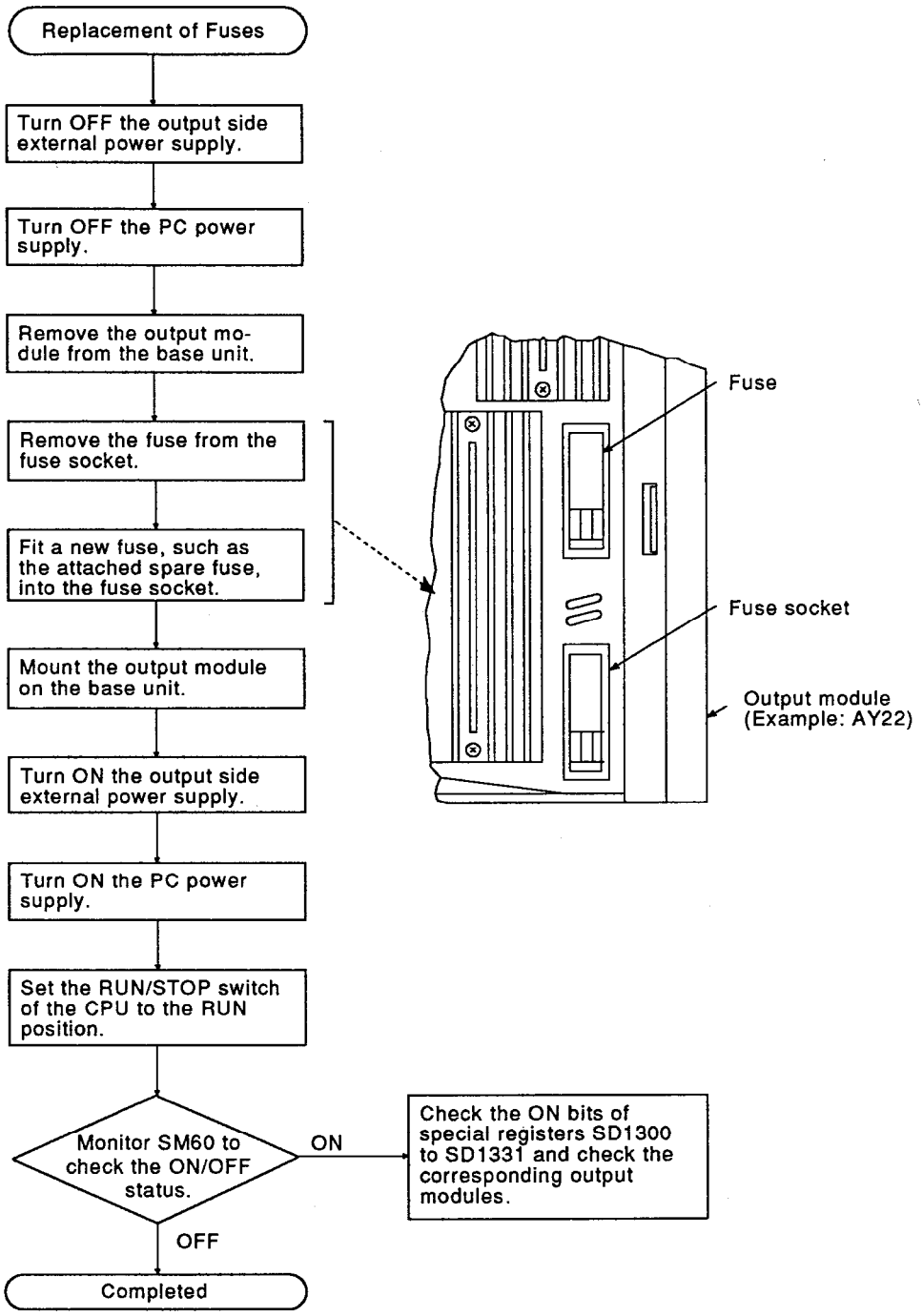
Even if the fuse does not blow, its element may have deteriorated due to rush current. Therefore, you are recommended to change the fuse periodically.

20.4.1 Replacing the power supply fuse

The procedure for replacing the fuse is as follows.



20.4.2 Replacing an output module fuse



MEMO

21. TROUBLESHOOTING

This chapter describes the various errors that may occur while using the system, how to determine the causes of errors, and corrective actions to take when errors occur.

21.1 Troubleshooting Basics

System reliability depends not only on reliable equipment but also on short down-times in the event of trouble.
Troubleshooting is the process for identifying the cause of any trouble that occurs so that normal system operation can be restored quickly; there are three basic elements in this process, as indicated below.

(1) Visual checks

Check the following points.

(a) Machine motions (when stopped and when operating)

(b) Power ON/OFF status

(c) Status of I/O equipment

(d) Condition of wiring (I/O wires, cables)

(e) Display states of various indicators (such as POWER LED, RUN LED, ERROR LED, and I/O LED)

(f) Settings of setting switches (extension base, memory back up, etc.)

After checking (a) to (f), connect a peripheral device and check the operating state of the PC and the contents of the program.

(2) Trouble check

Perform the following operations and observe if there is any change in the trouble condition.

(a) Set the RUN/STOP key switch to STOP.

(b) Perform a reset operation using the RUN/STOP key switch.

(c) Turn the power ON and OFF.

(3) Narrow down the possible causes of the trouble.

From (1) and (2) above, deduce the location of the fault, i.e.:

(a) Inside or outside the PC?

(b) I/O module or other module?

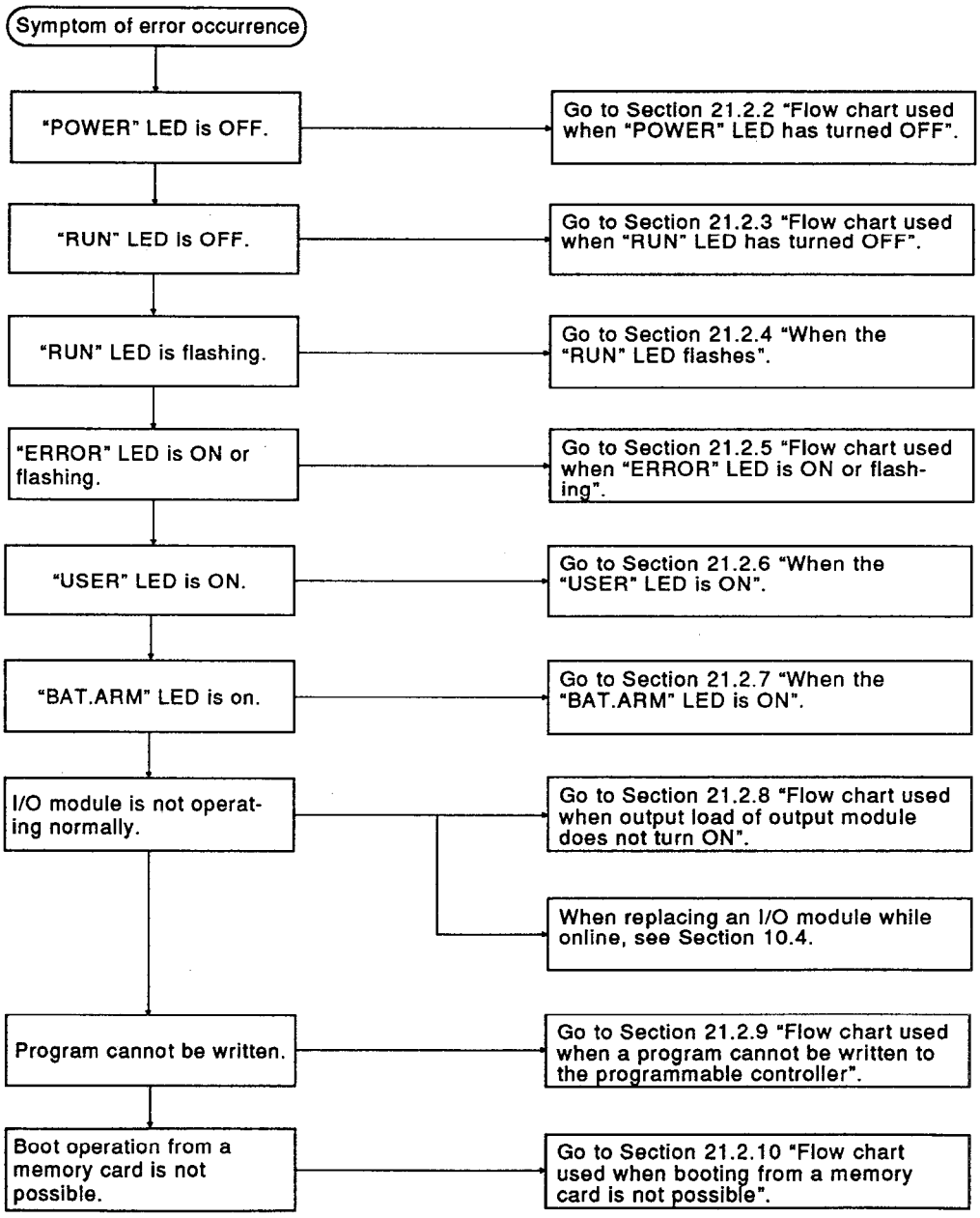
(c) Sequence program?

21.2 Troubleshooting

This section describes how to determine the nature of errors, the errors corresponding to each error code, and action to take when errors occur.

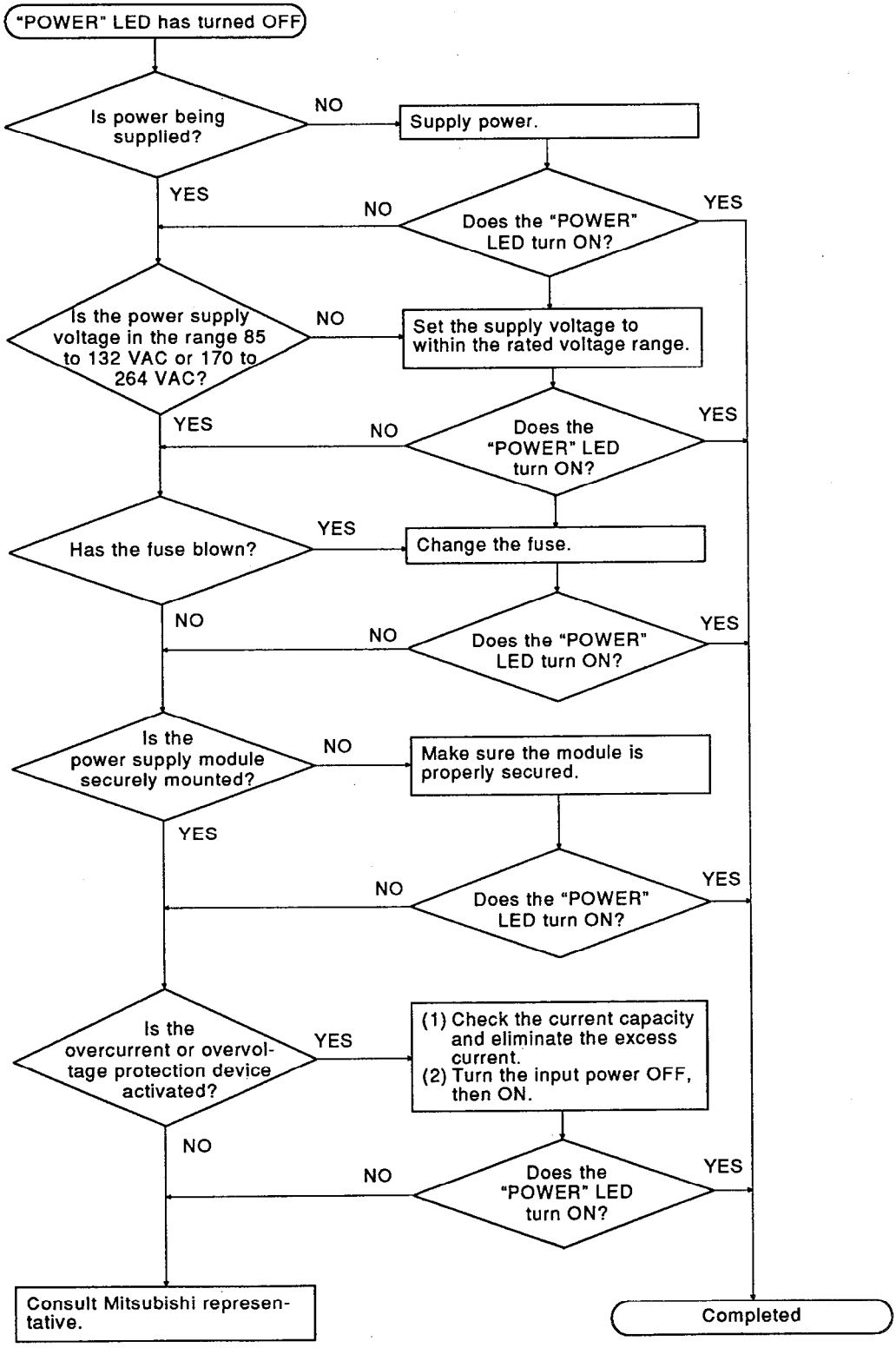
21.2.1 Troubleshooting flow chart

Errors are classified according to the associated symptoms and each is described in a different section, as follows.



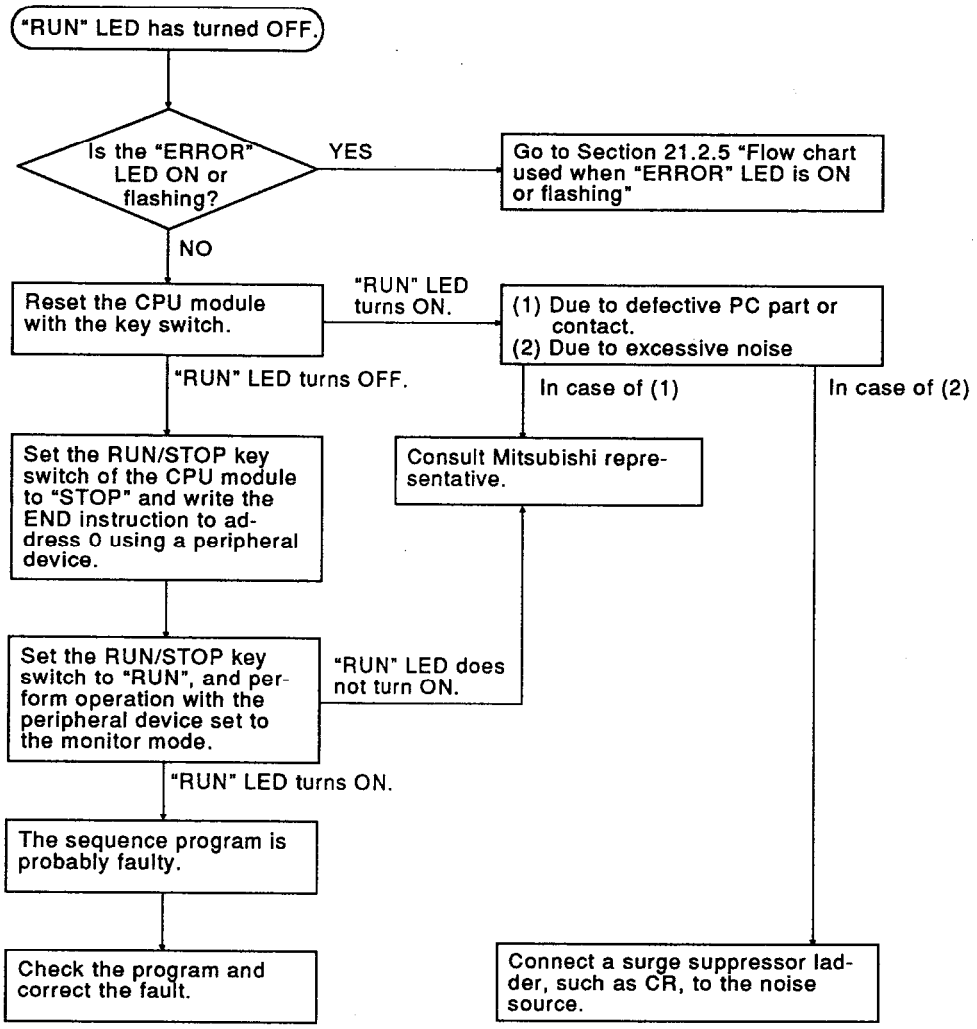
21.2.2 Flow chart used when "POWER" LED has turned OFF

The procedure to follow if the "POWER" LED is off after turning on the power or during operation is indicated in the flow chart below.



21.2.3 Flow chart used when "RUN" LED has turned OFF

The procedure to follow if the "RUN" LED turns off during operation is indicated in the flow chart below.

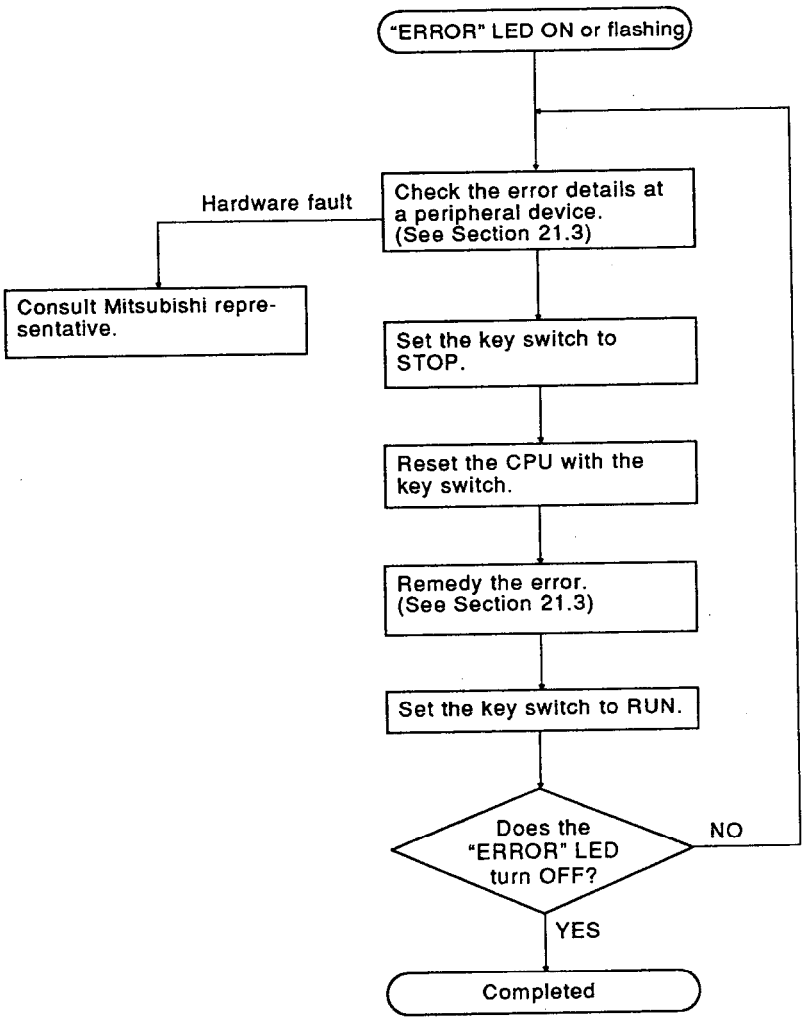


21.2.4 When the "RUN" LED flashes

The procedure to follow when the "RUN" LED flashes is described here.
With the QnACPU, when the key switch is turned from STOP to RUN after writing a program in the STOP state, the "RUN" LED flashes. In this state, there is no CPU error but CPU operation is stopped.
To set the CPU to the RUN state, either turn the key switch to STOP then RUN again, or reset the CPU using the key switch. The "RUN" LED will come ON.
With Q3ACPU and Q4ACPU, the message "PRG.CHECK!!" is displayed on the LED indicator on the front panel of the CPU module.

21.2.5 Flow chart used when "ERROR" LED is ON or flashing

The procedure to follow if the "ERROR" LED is ON or flashing when the power is turned ON, when operation is started, or during operation, is indicated in the flow chart below. Note that in the case of the Q3ACPU and Q4ACPU, an error message is displayed on the LED indicator on the front panel, and the nature of the error can be checked, and the appropriate action taken, by referring to the error code list in Section 21.4.



21.2.6 When the "USER" LED is ON

The procedure to follow when the "USER" LED turns ON is described here.
With the QnACPU, this LED comes ON when an error is detected by the CHK instruction, or when an annunciator, F, comes ON.
When this LED has come ON, monitor the special relays and special registers in the monitor mode at a peripheral device, and check if the CHK instruction or an annunciator, F, has come ON. (CHK instruction → SM80, SD80; annunciators, F → SM62, SD62 to SD79).
After checking, eliminate the reason that the "USER" LED came ON, and then turn the LED OFF either by resetting with the key switch or by executing an LEDR instruction.
With Q3ACPU and Q4ACPU, the LED annunciator on the front panel of the CPU module displays an error message or annunciator comment.

REMARK

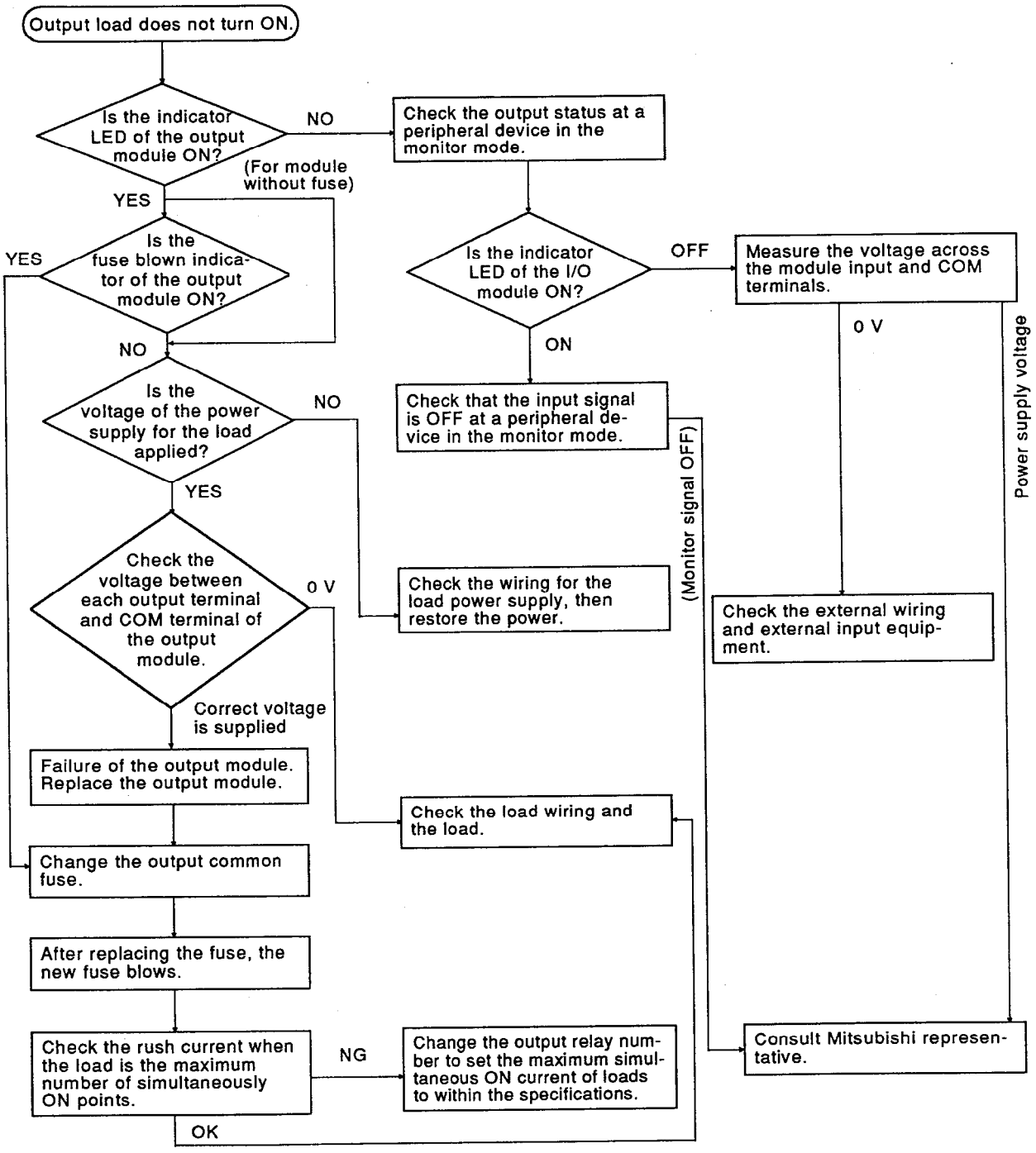
When the key switch is turned 3 times to "L.CLR" in a latch clear operation, the "USER" LED flashes to indicate that latch clear processing is in progress.
(With Q3ACPU and Q4ACPU, the message "L.CLR RDY" is displayed on the LED indicator on the front panel of the CPU module.)
When the key switch is turned one more time to "L.CLR" while the "USER" LED is flashing, the "USER" LED goes OFF and latch clear processing is ended.

21.2.7 When the "BAT.ARM" LED is ON

The procedure to follow when the "BAT.ARM" LED turns ON is described here.
With QnACPU, this LED turns ON when the voltage of the CPU module battery or the battery of a memory card drops.
When this LED has come ON, monitor the special relays and special registers in the monitor mode at a peripheral device, and check if there has been a voltage drop at either the CPU battery or a memory card battery (SM51 and SM52, SD51 and SD52).
After checking, replace the battery whose voltage has dropped with a new one, and then turn the LED OFF either by resetting with the key switch or by executing an LEDR instruction.
With Q3ACPU and Q4ACPU, the LED annunciator on the front panel of the CPU module displays a message.

21.2.8 Flow chart used when output load of output module does not turn ON

The procedure to follow if the output load of an output module does not turn ON during operation is indicated in the flow chart below.

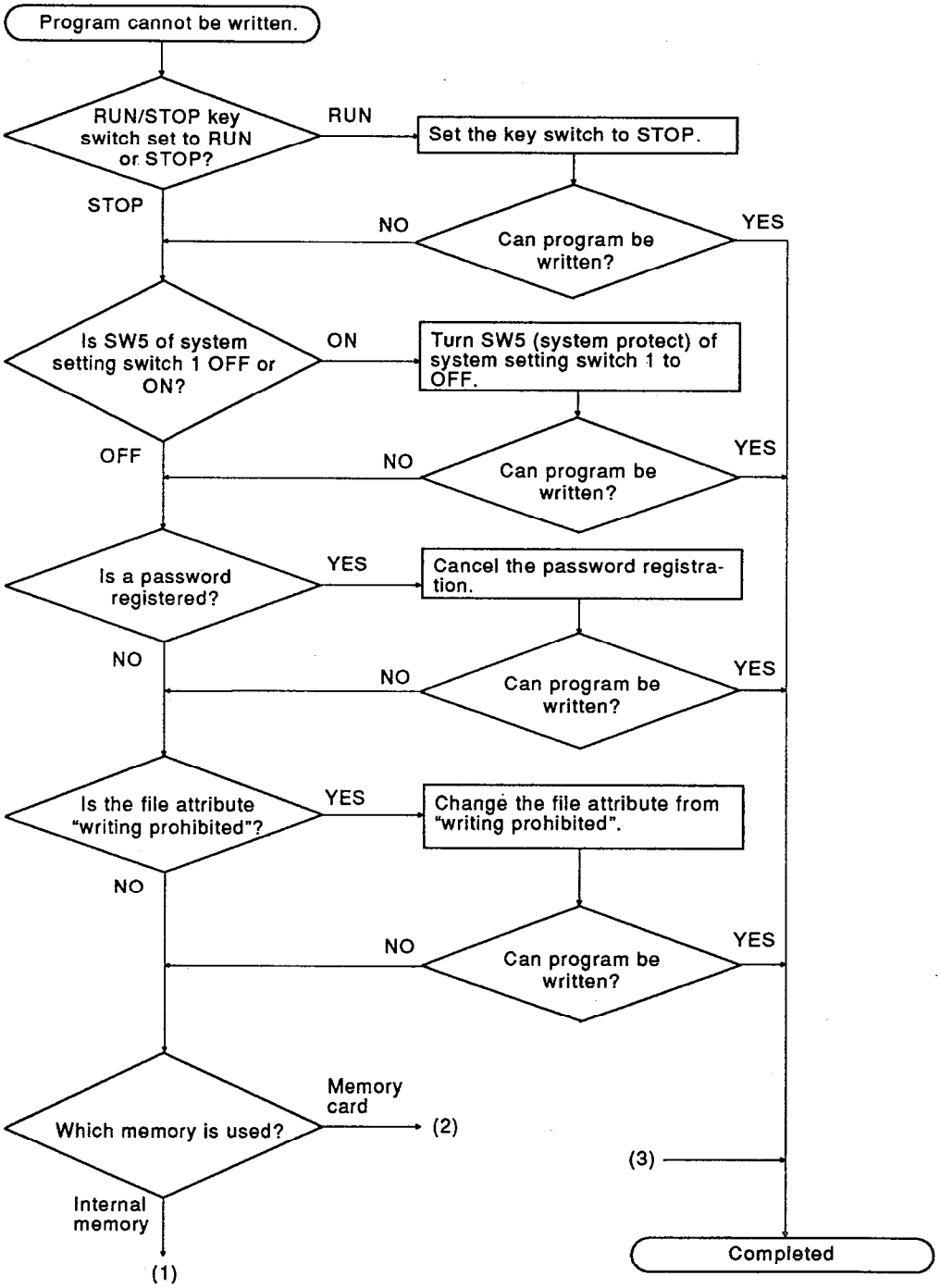


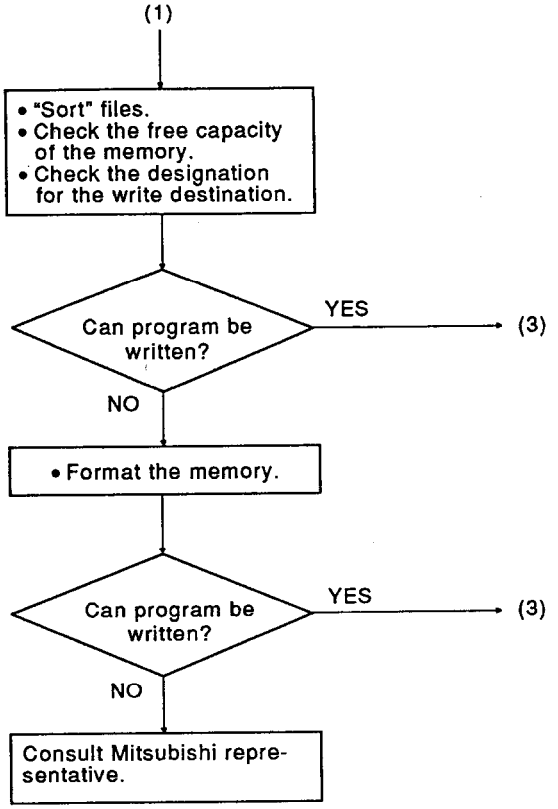
POINT

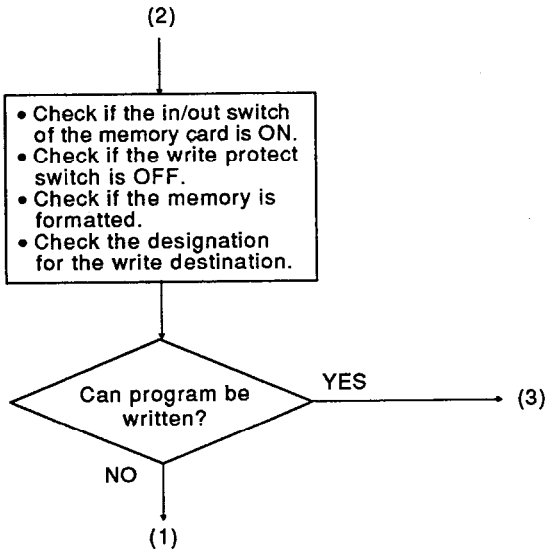
In the case of trouble where an input signal of an input module will not turn OFF, or a load of an output module will not turn OFF, carry out troubleshooting by referring to Section 21.4 "I/O Connection Troubleshooting".

21.2.9 Flow chart used when a program cannot be written to the programmable controller

The procedure to follow if an attempt to write programs etc. to the CPU fails is indicated in the flow chart below.

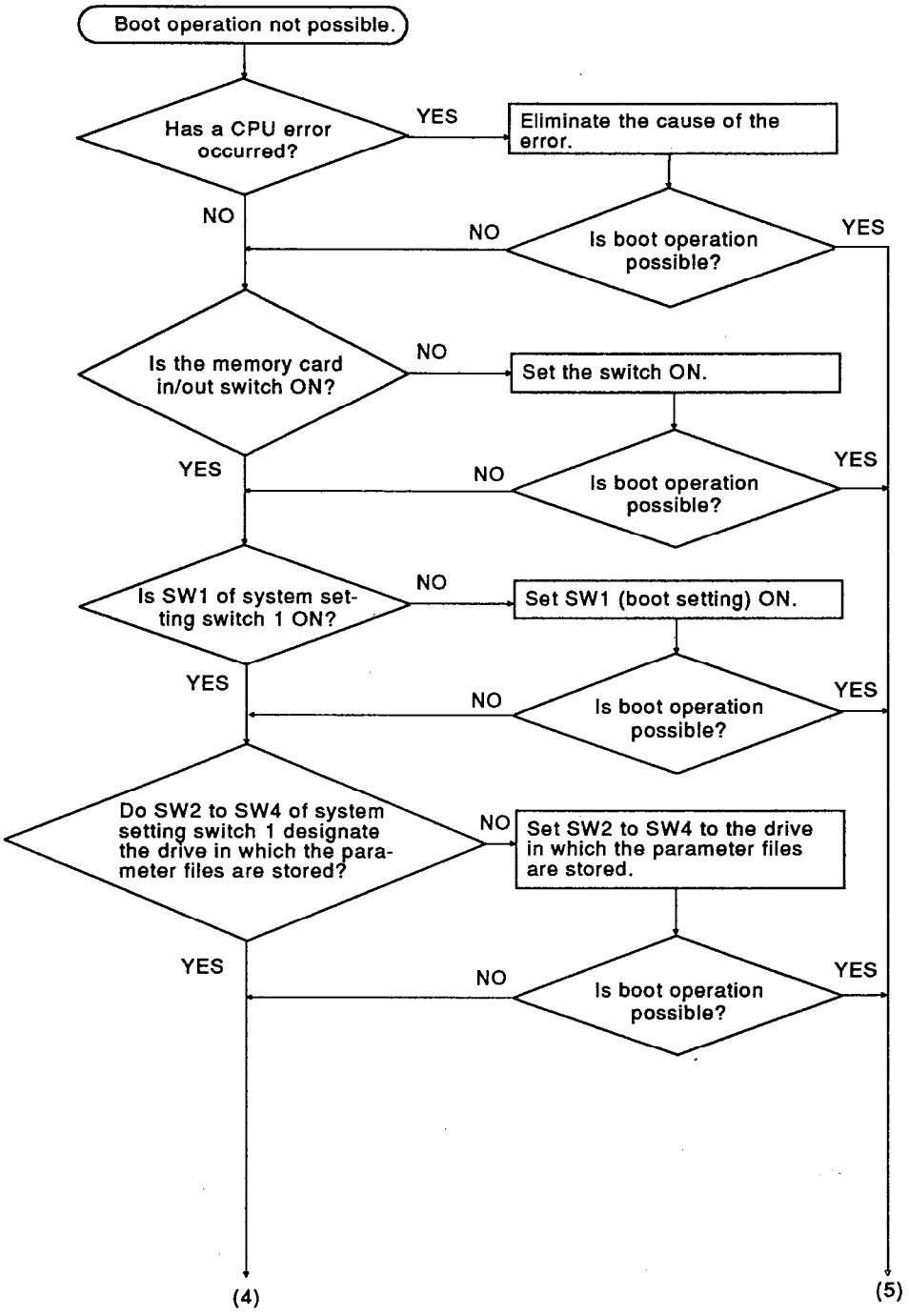


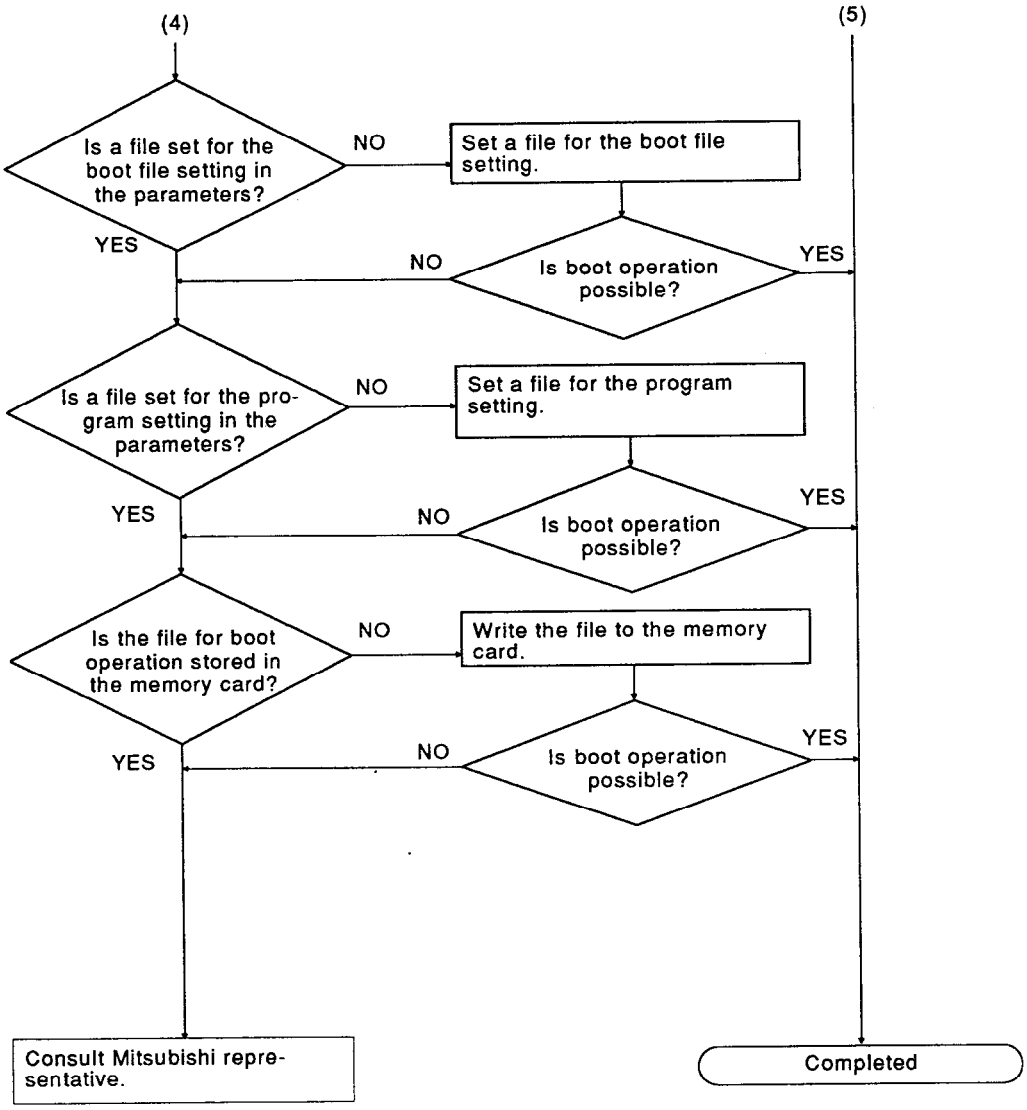

(1)
• "Sort" files.
• Check the free capacity of the memory.
• Check the designation for the write destination.
Can program be written?
YES
(3)
NO
• Format the memory.
Can program be written?
YES
(3)
NO
Consult Mitsubishi representative.


(2)
• Check if the in/out switch of the memory card is ON.
• Check if the write protect switch is OFF.
• Check if the memory is formatted.
• Check the designation for the write destination.
Can program be written?
YES
(3)
NO
(1)

21.2.10 Flow chart used when booting from a memory card is not possible

The procedure to follow if an attempt to boot the CPU from a memory card fails is indicated in the flow chart below.




(4)
(5)
Is a file set for the boot file setting in the parameters?
NO
Set a file for the boot file setting.
YES
NO
Is boot operation possible?
YES
Is a file set for the program setting in the parameters?
NO
Set a file for the program setting.
YES
NO
Is boot operation possible?
YES
Is the file for boot operation stored in the memory card?
NO
Write the file to the memory card.
YES
NO
Is boot operation possible?
YES
Consult Mitsubishi representative.
Completed

21.3 Error Codes

When a programmable controller error occurs at power ON, on switching to the RUN state, or during the RUN state, the self-diagnosis function displays the error content (by LED indication, or message display on an LED indicator), and stores the error information at a special relay (SM) and special register (SD).
QnACPU errors and corrective actions are described in this section.

21.3.1 Error code readout procedure

When an error occurs, the error code or error message, etc., can be read out using GPPQ.
For details regarding the GPPQ operation procedure, refer to the SW□IVD-GPPQ Operating Manual (Online).

21.3.2 Error codes

Error code/message descriptions, causes, and corrective actions are shown below.

Error Code (SD0)*1	Error Message	Common Information (SD5 to 12) *1	Individual Information (SD13 to 20) *1	LED Status		CPU Operating State	Diagnosis Timing
				RUN	ERROR		
1000	MAIN CPU DOWN	—	—	OFF	Flashing	Stop	Constant check
1010 1011 1012	END NOT EXECUTE	—	—	OFF	Flashing	Stop	When an END instruction is executed
1101 1102 1103 1104	RAM ERROR	—	—	OFF	Flashing	Stop	At power ON & reset
1200 1201 1202	OPE. CIRCUIT ERR.	—	—	OFF	Flashing	Stop	At power ON & reset
1300	FUSE BREAK OFF	Module No.	—	OFF/ON	Flashing /ON	Stop /Operation continues *2	When an END instruction is executed
1310	I/O INT ERROR	Module No.	—	OFF	Flashing	Stop	At interruption
1401	SP. UNIT DOWN	Module No.	Program error location	OFF	Flashing	Stop	At power ON & reset
1402							When a FROM/TO instruction is executed
1411	CONTROL-BUS ERR.	Module No.	Program error location	OFF	Flashing	Stop	At power ON & reset
1412							When a FROM/TO instruction is executed
1500	AC DOWN	—	—	ON	OFF	Operation continues	Constant check
1600 1601 1602	BATTERY ERROR	Drive name	—	ON BAT. ALM LED flashing	OFF	Operation continues	Constant check
2000	UNIT VERIFY ERR.	Module No.	—	OFF /ON	Flashing /ON	Stop /Operation continues *2	When an END instruction is executed
2100 2101 2102 2103 2104	SP. UNIT LAY ERR.	Module No.	—	OFF	Flashing	Stop	At power ON & reset

*1 Values shown in parentheses indicate the special register numbers where the error information is stored.
*2 The CPU operating state when an error occurs can be designated by parameter setting. (LED operation changes accordingly.)

	Error Description & Cause	Corrective Action
	Main CPU run-away or failure. (1) Malfunction caused by noise, etc. (2) Hardware faults	(1) Take noise-prevention measures. (2) Reset, then set to RUN again. If the same error occurs again, the problem is probably a CPU hardware fault. In this case, contact your service representative for assistance.
	The END instructions are ignored and all the programs in the total program capacity are executed. (1) The END instruction is being read as another instruction due to noise, etc. (2) The END instruction has been changed to another instruction somehow.	(1) Take noise-prevention measures. (2) Reset, then set to RUN again. If the same error occurs again, the problem is probably a CPU hardware fault. In this case, contact your service representative for assistance.
	Fault at built-in RAM for CPU sequence program storage.	As the problem is a CPU hardware fault, contact your service representative for assistance.
	Fault at CPU's internal work area RAM.	
	Fault at CPU's internal device memory.	
	Fault at CPU's internal address RAM.	
	Fault at operation ladder for index qualifications in the CPU.	
	Fault at CPU's internal hardware (logic).	
	Fault at operation ladder that executes sequence processing in the CPU.	
	A fuse is blown at one of the output modules.	(1) Check the blown-fuse LED indicators for the output modules, and replace the blown fuse. (2) Read out the error common information at a peripheral device, then replace the fuse at the output module corresponding to that information (module No.). Or, monitor special registers SD1300 and SD1331 at a peripheral device, and replace the fuse at the output module whose bit setting is "1".
	An interruption occurs even though no interrupt module is installed.	There is a hardware fault at one of the modules. Contact your service representative for assistance.
	When I/O allocations have been made in the parameters, there is no reply from the special function module when the initial communication occurs. When the error occurs, the head I/O No. of the special function module indicated by the common information is stored.	There is a hardware fault at the accessed special function module. Contact you service representative for assistance.
	The special function module was accessed on execution of a FROM/TO instruction, but there was no reply. When the error occurs, the program error location indicated by the individual information is stored.	
	When I/O allocations have been made in the parameters, a special function module cannot be accessed at the initial communication. When the error occurs, the head I/O No. of the special function module indicated by the common information is stored.	There is a fault at the special function module, the CPU, or the base unit. Contact you service representative for assistance.
	Execution of a FROM/TO instruction is prevented by a control bus error with the special function module. When the error occurs, the program error location indicated by the individual information is stored.	
	A momentary power interruption occurred.	Check the power supply.
	(1) Voltage of the CPU battery fell below the prescribed level. (2) A CPU battery lead connector is not installed.	(1) Replace the battery. (2) Install a lead connector at systems where a built-in RAM or memory backup function is used.
	Voltage of the memory card 1 battery fell below the prescribed level.	Replace the battery.
	Voltage of the memory card 2 battery fell below the prescribed level.	
	The input/output module configuration differs from the one regsitered at power ON. • An input/output module (including special function modules) is either loose or disconnected during operation.	Read out the error common information to a peripheral device, then check/replace the module corresponding to that information (module No.). Or, monitor special registers SD1400 to SD1431 at the peripheral device, and replace the module whose bit setting is "1".
	An input/output module position has been allocated to a special function module (or vice-versa) in the parameter I/O allocations.	Discrepancy with input/output module information at power ON.
	The number of special function modules (excluding AI61) which can designate an interrupt start at the CPU exceeds 8 modules.	Install 8 or fewer special function modules which can designate an interrupt start. (Except for AI61)
	The number of installed serial communication modules, etc., exceeds 7 modules.	Install 6 or fewer serial communication modules, etc.
	More than one AI61 interrupt modules is installed.	Install only one AI61 module.
	A discrepancy exists between the module allocations designated in MELSECNET/MINI auto-refresh parameter setting and the names of modules which are actually linked.	Correct the module allocations designated by the MELSECNET/MINI auto-refresh setting to conform with the names of modules which are actually linked.

Error Code List (cont.)

Error Code (SD0)	Error Message	Common Information (SD5 to 12)	Individual Information (SD13 to 20)	LED Status		CPU Operating State	Diagnosis Timing
				RUN	ERROR		
2105	SP. UNIT LAY ERR.	Module No.	—	OFF	Flashing	Stop	At power ON & reset
2106							
2107							
2108							
2110	SP. UNIT ERROR	Module No.	Program error location	OFF /ON	Flashing /ON	Stop /Operation continues *2	When a FROM/TO instruction is executed
2111							
2112							
2113		FFFFH (fixed)					
2200	MISSING PARA.	Drive name	—	OFF	Flashing	Stop	At power ON & reset
2210	BOOT ERROR	Drive name	—	OFF	Flashing	Stop	At power ON & reset
2300	ICM. OPE. ERROR	Drive name	—	OFF /ON	Flashing /ON	Stop /Operation continues *2	When memory card is inserted/removed
2301							
2302							
2400	FILE SET ERROR	File name	Parameter No.	OFF	Flashing	Stop	At power ON & reset
2401							
2410	FILE OPE. ERROR	File name	Program error location	OFF /ON	Flashing /ON	Stop /Operation continues *2	When an instruction is executed
2411							
2412							
2413							
2500	CAN'T EXE. PRG.	File name	—	OFF	Flashing	Stop	At power ON & reset
2501							
2502							
2503							
2504							

*2 The CPU operating state when an error occurs can be designated by parameter setting. (LED operation changes accordingly.)

Error Description & Cause	Corrective Action
The number of registered I/O allocations for special function modules which can use dedicated instructions exceeds the maximum limit. (The number, as calculated below exceeds 1344) (AD59 number installed x 5) (AD57(S1)/AD58 number installed x 8) (AJ71C24(S3/S6/S8) number installed x 10) (AJ71UC24 number installed x 10) (AJ71C21(S1) number installed x 29) + (AJ71PT32(S3) number installed x 125) Total >1344	Reduce the number of installed special function modules.
(1) 5 or more AJ71QLP21, AJ71QBR11 modules are installed. (2) 3 or more AJ71AP21/R21, AJ71AT21B modules are installed. (3) The total number of installed AJ71QLP21, AJ71QBR11, AJ71AP21/R21, and AJ71AT21B modules exceeds 4. (4) Identical network Nos. or identical station Nos. exist at the MELSECNET/10 network system. (5) 2 or more master or local stations exist simultaneously at the MELSECNET (II) or MELSECNET/B data link system.	(1) Install 4 or fewer modules. (2) Install 2 or fewer modules. (3) Reduce the total number of modules to 4 or less. (4) Check the network Nos. and station Nos. (5) Check the station Nos.
The first X/Y in the parameter I/O allocation settings duplicates the first X/Y of another module.	Discrepancy with input/output module information at power ON.
An AJ71LP21 or AJ71BR11 network module for use with AnUCPU has been installed.	Replace with AJ71QLP21 or AJ71QBR11 network modules.
No special function module existed when a FROM/TO instruction was executed.	Read out the error individual information and check/correct the FROM/TO instruction corresponding to that information (program error location).
No network module existed when a link direct device (J□\□) was designated.	
No special function module existed when a special function module dedicated instruction was executed. Or, no applicable special function module existed at that time.	Read out the error individual information and check/correct the special function module dedicated instruction corresponding to that information (program error location).
No simulation special function module data was designated in the simulation data.	Read out the error individual information and designate simulation data for the special function module corresponding to that information (program error location).
No parameter file exists at the drive designated by the parameter enabled drive switch (DIP switch).	Check/correct the parameter enabled drive switch setting. Designate a parameter file for the drive designated by the parameter enabled drive switch.
Although the boot switch (DIP switch) is ON, there is no boot file (or the file data is incorrect) at the drive specified by the parameter enabled drive switch.	Check/correct the parameter enabled drive switch setting. Set a boot file at the drive designated by the parameter enabled drive switch. Check/correct the boot file data.
A memory card was removed without turning the memory card in/out switch ON.	Remove the memory card after turning the memory card in/out switch ON.
(1) Memory card is not formatted. (2) Memory card format is incorrect.	(1) Format the memory card. (2) Re-format the memory card.
A memory card which cannot be used with QnACPU was inserted.	Check the memory card.
The file specified in "PC file setting" parameter setting does not exist.	Read out the error individual information to a peripheral device, and check/correct the drive name and file name corresponding to that information (parameter No.). Create the specified file.
File designated for the fault history file (PC RAS parameter setting), etc., cannot be created.	Read out the error individual information to a peripheral device, and check/correct the drive name and file name corresponding to that information (parameter No.). Check the memory card's remaining memory area.
The file designated by the sequence program does not exist.	Read out the error individual information to a peripheral device, and check/correct the program corresponding to that information (program error location). Create the specified file.
An off-limits file (comment file, etc.) was designated by the sequence program.	Read out the error individual information to a peripheral device, and check/correct the program corresponding to that information (program error location).
An off-limits SFC program file was designated by the sequence program.	Read out the error individual information to a peripheral device, and check/correct the program corresponding to that information (program error location).
No data could be written to a file designated by the sequence program.	Read out the error individual information to a peripheral device, and check/correct the program corresponding to that information (program error location). Check to see if the designated file is write-protected.
A program file exists with a device outside the device allocation range designated by the device settings in the parameters.	Read out the error common information to a peripheral device, and check/correct the device allocations for the program file corresponding to that information (file name), and the device allocations in the parameter device settings.
Multiple program files exist even though none is designated in "program setting" parameter setting.	Change the "program setting" parameter to present. Or, delete unnecessary programs.
The program file is incompatible with the QnACPU. Or, the file content is not a sequence program.	Check that the program file type is "***.QPG". Check that the file content is a sequence program.
No program files exist.	Check the program configuration.
2 or more ordinary and control SFC programs were executed.	Check the parameter and program configuration

Error Code List (cont.)

Error Code (SD0)	Error Message	Common Information (SD5 to 12)	Individual Information (SD13 to 20)	LED Status		CPU Operating State	Diagnosis Timing
				RUN	ERROR		
3000 3001 3003 3004	PARAMETER ERROR	File name	Parameter No.	OFF	OFF	Stop	At power ON, reset, and STOP→RUN switching
3100 3101 3102	LINK PARA. ERROR	File name	Parameter No.	ON	ON	Operation continues	At power ON, reset, and STOP→RUN switching
3200 3201 3202 3203	SFC PARA. ERROR	File name	Parameter No.	OFF	Flashing	Stop	STOP → RUN
4000 4001 4003 4003 4004	INSTRCT CODE. ERR.	Program error location	—	OFF	Flashing	Stop	At power ON, reset, and STOP→RUN switching
4010	MISSING END INS.	Program error location	—	OFF	Flashing	Stop	At power ON, reset, and STOP→RUN switching
4020 4021	CAN'T SET (P)	Program error location	—	OFF	Flashing	Stop	At power ON, reset, and STOP→RUN switching
4030	CAN'T SET (I)	Program error location	—	OFF	Flashing	Stop	At power ON, reset, and STOP→RUN switching
4100 4101 4102 4103	OPERATION ERROR	Program error location	—	OFF/ON	Flashing /ON	Stop /Operation continues *2	When an instruction is executed
4200 4201 4202 4203	FOR NEXT ERROR	Program error location	—	OFF	Flashing	Stop	When an instruction is executed
4210 4211 4212 4213	CAN'T EXECUTE (P)	Program error location	—	OFF	Flashing	Stop	When an instruction is executed
4220 4221 4223	CAN'T EXECUTE (I)	Program error location	—	OFF	Flashing	Stop	When an instruction is executed
4230 4231 4232 4233 4234 4235 4236 4237 4238	INST. FORMAT ERR.	Program error location	—	OFF	Flashing	Stop	When an instruction is executed

*2 The CPU operating state when an error occurs can be designated by parameter setting. (LED operation changes accordingly.)

	Error Description & Cause	Corrective Action
	Settings for the following parameter items are outside the permissible range for CPU use: timer limit setting, RUN-PAUSE contact, common pointer No., general data processing, number of vacant slot points, system interrupt setting.	(1) Read out the error detailed information to a peripheral device, and check/correct the parameter item corresponding to that information (parameter No.). (2) If the error occurs again after the parameter setting has been corrected, there is probably a fault at the CPU's internal memory, or at the memory card. Contact your service representative for assistance.
	Parameter data has been destroyed.	
	The number of devices designated at the device settings in the parameters is outside the permissible range for CPU use.	
	The parameter file is incompatible with the QnACPU. Or, the file does not contain parameter data.	Check that the parameter file type is "***. QPA". Check that the file contains parameter data.
	Although the QnACPU is the control or master station, no network parameter data has been written to it.	(1) Correct, then write the network parameters. (2) If the error occurs again after corrections are made, the problem is probably caused by a hardware fault. Contact your service representative for assistance.
	The refresh parameter data is incorrect.	
	An error occurred when the network parameters were checked at a network module.	
	Parameter data is incorrect.	Correct, then write the parameter data.
	SFC block attribute information is incorrect.	
	The number of step relays designated by parameter setting is fewer than the number used by the program.	
	The SFC program execution type designated by parameter setting is other than scan or standby".	
	The program contains an instruction code which cannot be decoded by the CPU.	Read out the error common information to a peripheral device, and check/correct the error step corresponding to that information (program error location).
	The program contains an SFC program dedicated instruction even though it is not an SFC program.	
	The instruction name of the extend instruction designated in the program is incorrect.	
	The number of devices of the extend instruction designated in the program is incorrect.	
	An unusable device has been designated for the extend instruction designated by the program.	
	The program contains no END (FEND) instruction.	Read out the error common information to a peripheral device, and check/correct the file corresponding to that information (program error location).
	The total number of pointers used in the program files exceeds the maximum allowable number designated by the parameter setting.	Read out the error common information to a peripheral device, and check/correct the error step corresponding to that information (program error location).
	Duplicate common pointer Nos. exist in the files.	
	Duplicate allocation pointer Nos. exist in the files.	Read out the error common information to a peripheral device, and check/correct the error step corresponding to that information (program error location).
	An instruction contains data which cannot be processed.	Read out the error common information to a peripheral device, and check/correct the error step corresponding to that information (program error location).
	The instruction data exceeds the allowable number of data for instruction processing. Or, the storage data or constants at the device specified by the instruction exceeds the usable range.	
	Incorrect network No. or station No. designated with a network dedicated instruction.	
	Configuration of PID dedicated instruction is incorrect.	
	A FOR instruction was executed without a NEXT instruction. Or, the number of NEXT instructions is fewer than the number of FOR instructions.	Read out the error common information to a peripheral device, and check/correct the error step corresponding to that information (program error location).
	A NEXT instruction was executed when there was no FOR instruction. Or, the number of NEXT instructions is greater than the number of FOR instructions.	
	The nesting count exceeded 16 nestings.	Reduce the nesting count to 16 or less.
	A BREAK instruction was executed when there was no FOR instruction.	Read out the error common information to a peripheral device, and check/correct the error step corresponding to that information (program error location).
	A CALL instruction was executed without a destination pointer.	Read out the error common information to a peripheral device, and check/correct the error step corresponding to that information (program error location).
	The executed sub-routine program contained no RET instruction.	
	A RET instruction is required before the FEND instruction in the main program.	
	The nesting count exceeded 16 nestings.	Reduce the nesting count to 16 or less.
	An interruption input occurred without a corresponding interrupt pointer.	Read out the error common information to a peripheral device, and check/correct the error step corresponding to that information (program error location).
	The executed interrupt program contained no IRET instruction.	
	A IRET instruction is required before the FEND instruction in the main program.	
	The numbers of CHK and CHKEND instructions do not match.	Read out the error common information to a peripheral device, and check/correct the error step corresponding to that information (program error location).
	The numbers of IX and IXEND instructions do not match.	
	The configuration between the FOR and NEXT instructions at the ladder's extend instruction is incorrect.	
	The configuration between the DO and WHILE instructions at the ladder's extend instruction is incorrect.	
	The configuration between the SELECT and CASE instructions at the ladder's extend instruction is incorrect.	
	The CHK instruction's check conditions are incorrect. Or, a CHK instruction has been used in a low-speed execution program.	
	The number of ladder extend instruction nestings exceeded 16.	Reduce the nesting count to 16 or less.
	An EXITFOR instruction was executed when there was no FOR instruction (ladder extend instruction).	Read out the error common information to a peripheral device, and check/correct the error step corresponding to that information (program error location).
	An EXITDO instruction was executed when there was no DO instruction (ladder extend instruction).	

Error Codes (cont.)

Error Code (SD0)	Error Message	Common Information (SD5 to 12)	Individual Information (SD13 to 20)	LED Status		CPU Operating State	Diagnosis Timing
				RUN	ERROR		
4300 4301	EXTENDINST. ERR.	Program error location	—	OFF /ON	Flashing /ON	Stop /Operation continues *2	When an instruction is executed
4400	SFCP. CODE ERROR	Program error location	—	OFF	Flashing	Stop	STOP → RUN
4410 4411	CAN'T SET (BL)	Program error location	—	OFF	Flashing	Stop	STOP → RUN
4420 4421 4422	CAN'T SET (S)	Program error location	—	OFF	Flashing	Stop	STOP → RUN
4500 4501 4502 4503 4504	SFCP. FORMAT ERR.	Program error location	—	OFF	Flashing	Stop	STOP → RUN
4600 4601 4602	SFCP. OPE. ERROR	Program error location	—	OFF/ON	Flashing /ON	Stop /Operation continues *2	When an instruction is executed
4610 4611	SFCP. EXE. ERROR	Program error location	—	ON	ON	Operation continues	STOP → RUN
4620 4621	BLOCK EXE. ERROR	Program error location	—	OFF	Flashing	Stop	When an instruction is executed
4630 4631 4632 4633	STEP EXE. ERROR	Program error location	—	OFF	Flashing	Stop	When an instruction is executed
5000 5001	WDT ERROR	Time period (setting value)	Time period (actually measured value)	OFF	Flashing	Stop	Constant check
5010 5011	PRG. TIME OVER	Time period (setting value)	Time period (actually measured value)	ON	ON	Operation continues	Constant check
9000	F**** *3	Program error location	Annunciator No.	ON / OFF USER LED ON		Operation continues	When an instruction is executed
9010	<CHK>ERR***-*** *4	Program error location	Fault No.	ON / OFF USER LED ON		Operation continues	When an instruction is executed

*2 The CPU operating state when an error occurs can be designated by parameter setting. (LED operation changes accordingly.)
*3 The detected annunciator number is displayed at ****
*4 The detected contact and coil numbers are displayed at ***

	Error Description & Cause	Corrective Action
	An incorrect MELSECNET/MINI-S3 master module control instruction was designated.	Read out the error common information to a peripheral device, and check/correct the error step corresponding to that information (program error location).
	An incorrect AD57/AD58 control instruction was designated.	
	The SFC program contains no SFCP and SFCPEND instructions.	Read out the error common information to a peripheral device, and check/correct the error step corresponding to that information (program error location).
	The block No. designated by the SFC program exceeds the maximum setting value.	
	Duplicate block Nos. are designated for the SFC program.	
	The number of step Nos. designated for the SFC program exceeds 255.	
	The total number of steps at all SFC programs exceeds the maximum setting value.	Reduce the total number of steps to fewer than the maximum value.
	Duplicate step Nos. are designated at the SFC program.	Read out the error common information to a peripheral device, and check/correct the error step corresponding to that information (program error location).
	The number of BLOCK and BEND instructions at the SFC program do not match.	
	The STEP∗ to TRAN∗ to TSET to SEND instruction configuration at the SFC program is incorrect.	
	There is no STEPI∗ instruction in an SFC program block.	
	The step designated by the SFC program's TSET instruction does not exist.	
	The step designated by the SFC program's TAND instruction does not exist.	
	The SFC program contains data which cannot be processed.	
	The device range which can be designated by the SFC program has been exceeded.	
	A block control END instruction preceded a START instruction at the SFC program.	
	Active step information was incorrect when an SFC program resumptive start occurred.	The program automatically executes an initial start.
	The key-switch was reset during the RUN state when "continue" was designated at the SFC program.	
	The SFC program designated a block start for a block which has already been started.	Read out the error common information to a peripheral device, and check/correct the error step corresponding to that information (program error location).
	The SFC program designated a block start for a block which does not exist.	
	The SFC program designated a step start for a step which has already been started.	
	The SFC program designated a step start for a step which does not exist.	
	The number of simultaneously active block steps which can be designated at the SFC program has been exceeded.	
	The total number of simultaneously active steps which can be designated has been exceeded.	
	The scan time at an initial execution type program exceeded the initial execution monitor period designated by the PC RAS parameter setting.	Read out the error individual information, and reduce the scan time in accordance with the information (time period).
	The program scan time exceeded the WDT value designated by the PC RAS parameter setting.	
	The program scan time exceeded the constant scan period designated by the PC RAS parameter setting.	
	The low-speed scan time exceeded the low-speed execution watchdog time designated by the PC RAS parameter setting.	
	Annunciator F switched ON.	Read out the error individual information, and check the program corresponding to that information (annunciator No.).
	An error was detected with the CHK instruction.	Read out the error individual information, and check the program corresponding to that information (fault No.).

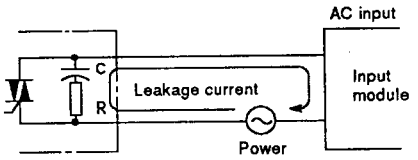
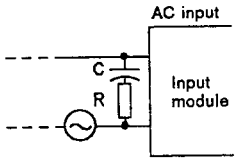
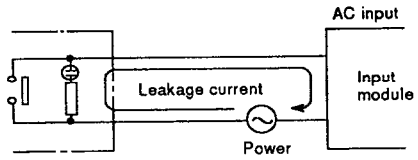
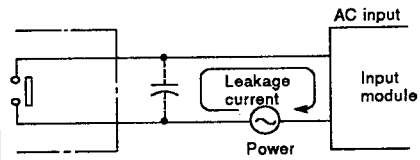
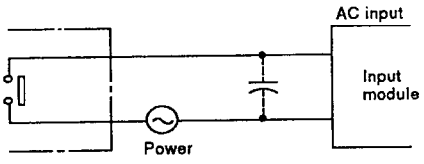
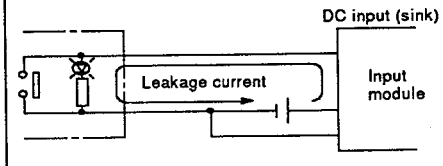
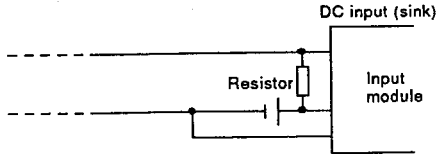
21.4 I/O Connection Troubleshooting

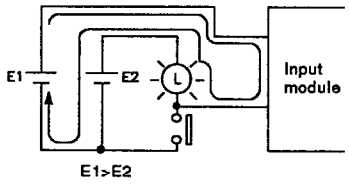
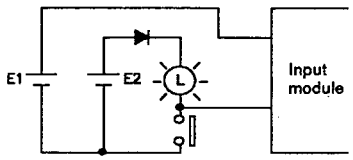
This section describes possible problems with I/O ladders and the corrective action to take in response to these problems.

21.4.1 Input ladder troubleshooting

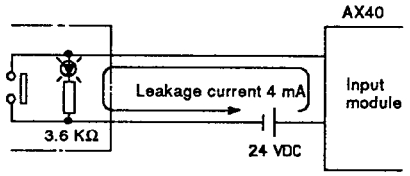
This section describes possible problems with the input ladder and corrective actions.

Input Ladder Troubleshooting

	Condition	Cause	Corrective Action
Example 1	Input signal does not turn OFF.	• Leakage current of input switch (e.g. drive by contactless switch) 	• Connect an appropriate resistor to make the voltage across the terminals of the input module lower than the OFF voltage.  A value in the following range is recommended for the CR constant: 0.1 to 0.47 μF + 47 to 120 Ω (1/2 W).
Example 2	Input signal does not turn OFF.	• Drive by a limit switch with neon lamp. 	• Same as Example 1. • Or make another, independent, display ladder.
Example 3	Input signal does not turn OFF.	• Leakage current due to line capacity of wiring cable. Line capacity C of twisted pair wire is approx. 100 PF/m. 	• Same as Example 1. • However, leakage current is not generated when the power supply is located on the input equipment side as shown below. 
Example 4	Input signal does not turn OFF.	• Drive by switch with LED indicator. 	• Connect a resistor which will make the voltage across the input module terminals and common higher than the OFF voltage, as shown below.  * An example of the calculation of the resistor value is given overpage.

	Condition	Cause	Corrective Action
Example 5	Input signal does not turn OFF.	• Sneak path due to the use of two power supplies. 	• Use only one power supply. • Connect a sneak path prevention diode. (See figure below). 

Example calculation for Example 4

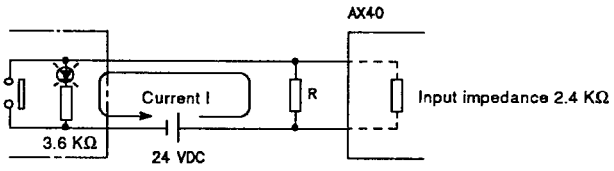


The switch with LED indicator is connected to an AX40 and there is 4 mA leakage current.

- The voltage V_{TB} across the terminal and common is obtained by the following expression:

$$V_{TB} = 4\ [mA] \times 2.4\ [K\Omega] = 9.6\ [V]$$ (the voltage drop of the LED is ignored)

Since this voltage does not satisfy the OFF voltage of 6 V or lower, the input signal does not turn OFF. Therefore, connect a resistance as shown below.



- Calculate the resistance value, R, as shown below:
In order to achieve an input voltage of less than 6 V, the current, I, must be:

$$(24 - 6\ [V]) \div 3.6\ [K\Omega] = 5\ mA$$

Accordingly, the resistance must be selected to give a current, I, of at least 5 mA.

- Hence, the resistance, R, must comply with the following:

$$6\ [V] \div R > 5 - 2.5\ [mA]$$
$$6\ [V] \div 2.5\ mA > R$$
$$2.4\ [K\Omega] > R$$

If a resistance, R, of 2 [KΩ] is used, the power capacity, W, must comply with the following:

$$W = (\text{applied voltage})^2/R \quad (\text{or, } W = (\text{maximum current})^2 \times R)$$

The terminal-to-terminal voltage for resistance R is a maximum of 26.4 V (24 V x 110 %) when the external switch is ON.
Therefore, the power capacity W of resistance R is:

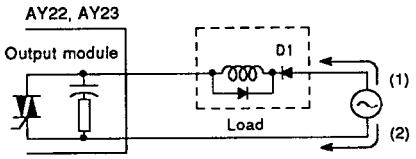
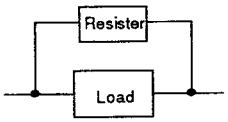
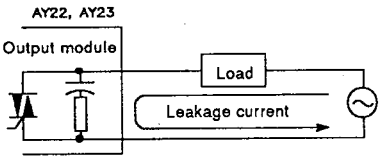
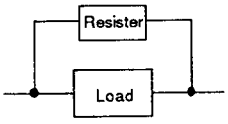
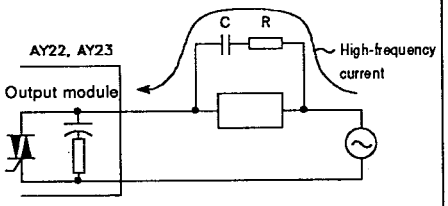
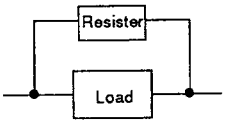
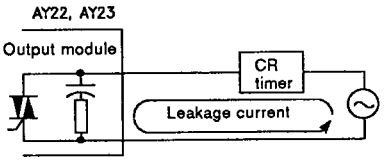
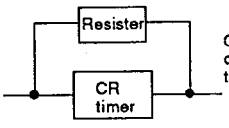
$$W = (26.4\ [V])^2/2[K\Omega] = 0.348\ [W]$$

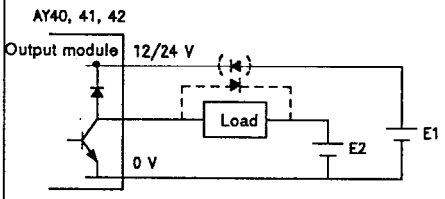
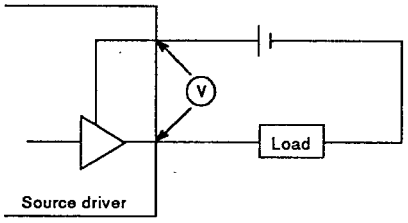
- Use a safety factor of 3 to 5. The resistor used should therefore have a rating of 1.0 to 1.7 [W].
In conclusion, a 2 [KΩ] , 1 to 2 [W] resistance should be connected between the problem terminal and its COM.

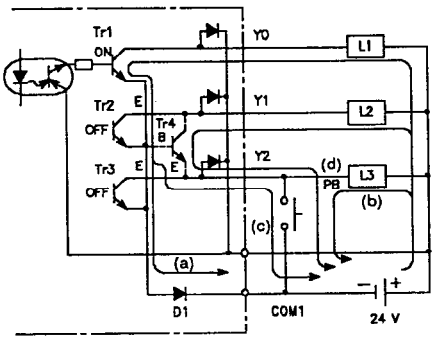
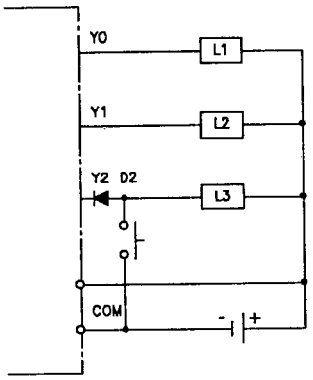
21.4.2 Output ladder troubleshooting

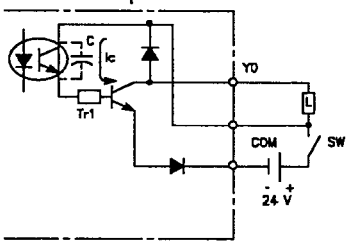
This section describes possible problems with the output ladder and corrective actions.

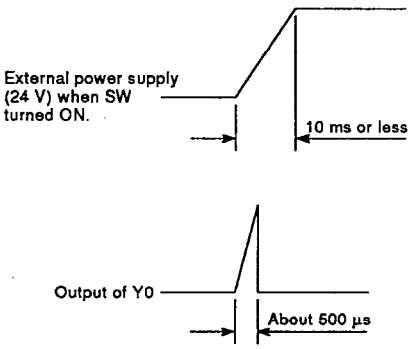
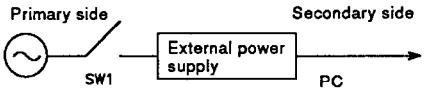
Output Circuit Troubleshooting

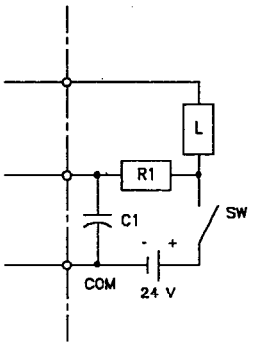
	Condition	Cause	Corrective Action
Example 1	When the output is OFF, excessive voltage is applied to the load.	• The load is half-wave rectified internally (seen in some solenoids).  • When the polarity of the power supply is as shown by (1), C is charged. When the polarity is as shown by (2), the voltage charged in C plus the power supply voltage are applied across D1. The maximum voltage is approximately 2.2 E.	• Connect a resistance of several tens or hundreds of KΩ across the load. (Using a resistor in this way poses no problem for the output device, but may cause deterioration of the diode incorporated in the load, resulting in burning.) 
Example 2	Load does not turn OFF (triac output).	• Leakage current due to built-in surge suppressor. 	• Connect the resistance across the terminals of the load. (When the wiring distance from the output module to the load is long, there may be a leakage current due to the line capacity.) 
Example 3	Load turns OFF with a delay (triac output).	• Leakage current due to surge suppressor for the load. 	• Disconnect the surge suppressor from between the load terminals, leaving only the resistance. (When the wiring distance from the output module to the load is long, there may be a leakage current due to the line capacity.)  Guide to resistance value: At 100 VAC: 5 to 10 KΩ, 5 to 3 W At 200 VAC: 10 to 20 KΩ, 15 to 10 W
Example 4	When the load is a CR type timer, the time constant fluctuates (triac output).		• Connect a resistance between the CR timer terminals. (When the wiring distance from the output module to the load is long, there may be a leakage current due to the line capacity.)  Calculate the resistance constant in accordance with the load.

	Condition	Cause	Corrective Action
Example 5	Load does not turn OFF. (Output by transistorwith clamp diode)	• Sneak path due to the use of two power supplies.  • Sneak path occurs when E1< E2.	• Use only one power supply. • Connect a sneak path prevention diode. (When the load is a relay or similardevice, a reverse voltage absorbingdiode must be connected to it. (shown by dotted line in figure at left)
Example 6	Load does not operate normally (e.g. due to external shorting). AY60EP AY80EP AY81EP AY82EP	External load malfunction or incorrect connection.	• Check the operation of the external load. • When output (Y) is ON, check the voltage between terminals as shown below. If the voltage is greater than 3 V, check the external load and wiring for short circuits. 

	Condition	Cause	Corrective Action
Example 7	When an external switch is connected in parallel between the output and common, the voltage across Y1 and COM1 drops to between 0 and 24 V even though the output Y1 which is not connected to the external switch is OFF. This output voltage problem occurs typically when the load L2 is relatively small (load current of several mA only), such as LED lamps and photocouplers. AY40 AY41 AY42	Faulty output by a parasitic transistor (Tr4).  Y2 can turn the load L3 on either from PC or PB. When PB is ON, Y0 is ON, and Y1 is OFF due to PC: (1) L1 (current (a)) and L3 (current (b)) turn ON. (2) A potential difference arises between the emitters, E, of Tr1 to 3 and COM1 since diode D1 is connected between E and COM1. (3) The transistors AY40 to 42, etc., are accompanied by a parasitic transistor (Tr4). (4) The potential difference mentioned in (2) above is supplied between the base (B) and emitter (E) of Tr4, causing the base current (c) to flow. (Tr4 turns ON). (5) The current in (4) causes the collector current to flow, and voltage Y1 drops to between 0 and 24 V.	 Add a diode D2 of the class I_F=1A to the output Y2 to connect an external switch as shown in the diagram above to prevent current from flowing into the lines marked (c) and (d) in the diagram to the left. However, check the operation voltage of L3 because the amount of voltage drop at Y2 at power ON increases for 0.6 to 1 V.

<table>
<tr><th></th><th>Condition</th><th>Cause</th><th>Corrective Action</th></tr>
<tr><td>Example 8</td><td>The load is momentarily turned ON when the external power supply rises. (Transistor output)</td><td>Faulty output due to the floating capacity (C) between the collector and emitter of the photocoupler.
(This is not a problem with normal loads, but in the case of highly sensitive loads (such as solid state relays), output faults are possible.)
Photocoupler
 

(1) If the external power supply rises sharply, current Ic flows due to the floating capacity (C) between the collector and emitter of the photocoupler.
(2) Current Ic flows to the base of the next stage transistor Tr1, and output Y0 turns ON for about 500 μs.
 
</td><td>(1) After checking that the external power supply takes at least 10 ms to rise when turned ON/OFF, set switch SW1 at the primary side of the external power supply.
 

(2) If it is necessary to set a switch at the secondary side of the external power supply, connect a capacitor and resistor to make the rise of the external power supply gentler (longer than 10 ms).
 

R1: Several tens of ohms
Power capacity ≥ (external power supply current)^{2} *1 x resistance value x (3 to 5) *2
C1: Several hundreds of μF, 50 mV
*1 Look up the current consumption of the external power supply of the module used in its manual.
*2 Select a resistor with a power capacity that is 3 to 5 times higher than the actual power consumption.
Example:
R1 = 40 Ω, C1 = 300 μF
Calculate the time constant as follows:
C1 x R1 = 300 x 10^{-6} x 40
=12 x 10^{-3}S
=12 ms</td></tr>
</table>

APPENDICES

APPENDIX 1 INSTRUCTION LISTS

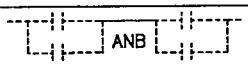
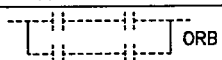
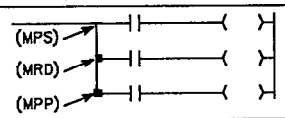
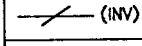
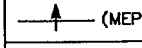
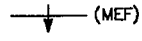
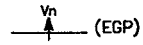
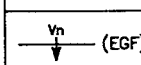
For details on SFC-related instructions, refer to the QnACPU Programming Manual (SFC).

1.1 Sequence Instructions

(1) Contact instructions

Classification	Symbol	Contents of Processing
Contact		• Logical operation start (A contact operation start)
		• Logical NOT operation start (B contact operation start)
		• Logical product (A contact series connection)
		• Logical product NOT (B contact series connection)
		• Logical add (A contact parallel connection)
		• Logical add NOT (B contact parallel connection)
		• Leading edge pulse operation start
		• Trailing edge pulse operation start
		• Leading edge pulse series connection
		• Trailing edge pulse series connection
		• Leading edge pulse parallel connection
		• Trailing edge pulse parallel connection

(2) Connection instructions

Classification	Symbol	Contents of Processing
Connection		• ANDs logical blocks (series connection of blocks)
		• ORs logical blocks (parallel connection of blocks)
		• Stores the operation result • Reads the operation result from MPS • Reads the operation result from MPS and clears the result
		• Inverts the operation result
		• Converts the operation result to a leading edge pulse
		• Converts the operation result to a trailing edge pulse
		• Converts the operation result to a leading edge pulse (stored at Vn)
		• Converts the operation result to a trailing edge pulse (stored at Vn)

(3) Output instructions

Classification	Symbol	Contents of Processing
OUT	—( )—	• Device output
	—[SET D]—	• Sets a device
	—[RST D]—	• Resets a device
	—[PLS D]—	• Generates one-program cycle pulse at the leading edge of an input signal
	—[PLF D]—	• Generates one-program cycle pulse at the trailing edge of an input signal
	—[FF D]—	• Inverts device output
	—[DELTA(P) D]—	• Converts direct output to pulses

(4) Shift instructions

Classification	Symbol	Contents of Processing
Shift	—[SFT(P) D]—	• Shifts a device 1 bit

(5) Master control instructions

Classification	Symbol	Contents of Processing
Master control	—[MC n D]—	• Master control start
	—[MCR n]—	• Master control reset

(6) Termination instructions

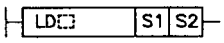
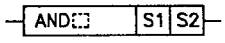
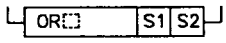
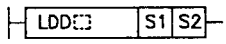
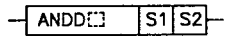
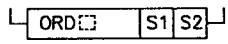
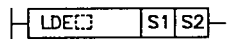
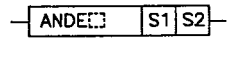
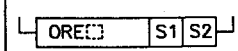
Classification	Symbol	Contents of Processing
Program end	—[FEND]—	• Terminates the main program
	—[END]—	• Terminates the sequence program

(7) Other instructions

Classification	Symbol	Contents of Processing
Stop	—[STOP]—	• Stops sequence operation on satisfaction of the input condition. • Sequence program execution can be resumed by turning the RUN/STOP key switch to RUN.
No processing	$\overline{\text{(NOP)}}$	• No processing (for program erasure or space)
	—[NOPLF]—	• No processing (for starting a new page during printout)
	—[PAGE n]—	• No processing (for managing the rest of the program as starting from step 0 of page "n")

1.2 Basic Instructions

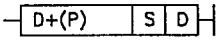
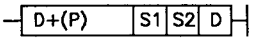
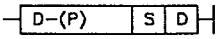
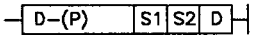
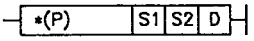
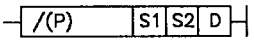
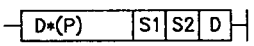
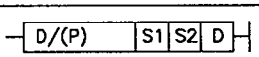
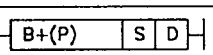
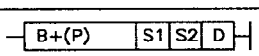
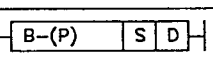
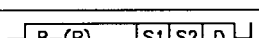
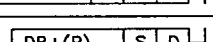
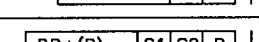
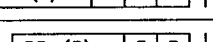
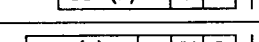
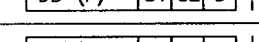
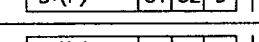
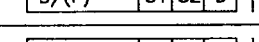
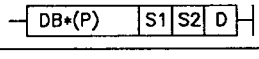
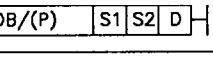
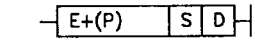
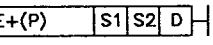
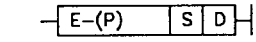
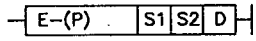
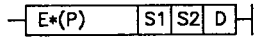
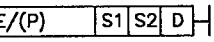
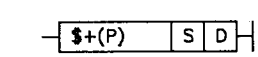
(1) Comparison instructions

Classification	Symbol	Contents of Processing
16-bit data comparison		[=] • Continuity when (S1) = (S2) • Non-continuity when (S1) ≠ (S2) [<>] • Continuity when (S1) ≠ (S2) • Non-continuity when (S1) = (S2)
		[>] • Continuity when (S1) > (S2) • Non-continuity when (S1) ≤ (S2) [<=] • Continuity when (S1) ≤ (S2) • Non-continuity when (S1) > (S2)
		[<] • Continuity when (S1) < (S2) • Non-continuity when (S1) ≥ (S2) [>=] • Continuity when (S1) ≥ (S2) • Non-continuity when (S1) < (S2)
32-bit data comparison		[=] • Continuity when (S1 + 1, S1) = (S2 + 1, S2) • Non-continuity when (S1 + 1, S1) ≠ (S2 + 1, S2) [<>] • Continuity when (S1 + 1, S1) ≠ (S2 + 1, S2) • Non-continuity when (S1 + 1, S1) = (S2 + 1, S2)
		[>] • Continuity when (S1 + 1, S1) > (S2 + 1, S2) • Non-continuity when (S1 + 1, S1) ≤ (S2 + 1, S2) [<=] • Continuity when (S1 + 1, S1) ≤ (S2 + 1, S2) • Non-continuity when (S1 + 1, S1) > (S2 + 1, S2)
		[<] • Continuity when (S1 + 1, S1) < (S2 + 1, S2) • Non-continuity when (S1 + 1, S1) ≥ (S2 + 1, S2) [>=] • Continuity when (S1 + 1, S1) ≥ (S2 + 1, S2) • Non-continuity when (S1 + 1, S1) < (S2 + 1, S2)
Real number data comparison		[=] • Continuity when (S1 + 1, S1) = (S2 + 1, S2) • Non-continuity when (S1 + 1, S1) ≠ (S2 + 1, S2) [<>] • Continuity when (S1 + 1, S1) ≠ (S2 + 1, S2) • Non-continuity when (S1 + 1, S1) = (S2 + 1, S2)
		[>] • Continuity when (S1 + 1, S1) > (S2 + 1, S2) • Non-continuity when (S1 + 1, S1) ≤ (S2 + 1, S2) [<=] • Continuity when (S1 + 1, S1) ≤ (S2 + 1, S2) • Non-continuity when (S1 + 1, S1) > (S2 + 1, S2)
		[<] • Continuity when (S1 + 1, S1) < (S2 + 1, S2) • Non-continuity when (S1 + 1, S1) ≥ (S2 + 1, S2) [>=] • Continuity when (S1 + 1, S1) ≥ (S2 + 1, S2) • Non-continuity when (S1 + 1, S1) < (S2 + 1, S2)

Classification	Symbol	Contents of Processing
Character string data comparison	LD$□ S1 S2	Character string S1 and character string S2 are compared character by character. Condition for "match": Character strings in which all characters match. Condition for "larger character string": Character string for which the character codes of the characters that are different are larger, or the character string which is longer. Condition for "smaller character string": Character string for which the character codes of the characters that are different are smaller, or the character string which is shorter. [=] • Continuity when (character string S1) = (character string S2) • Non-continuity when (character string S1) ≠ (character string S2) [>] • Continuity when (character string S1) > (character string S2) • Non-continuity when (character string S1) ≤ (character string S2) [<] • Continuity when (character string S1) < (character string S2) • Non-continuity when (character string S1) ≠ (character string S2) [>=] • Continuity when (character string S1) ≥ (character string S2) • Non-continuity when (character string S1) < (character string S2) [<=] • Continuity when (character string S1) ≤ (character string S2) • Non-continuity when (character string S1) > (character string S2)
	AND$□ S1 S2	
	OR$□ S1 S2	
Block data comparison	BKCMP=(P) S1 S2 D n BKCMP<>(P) S1 S2 D n BKCMP>(P) S1 S2 D n BKCMP<=(P) S1 S2 D n BKCMP<(P) S1 S2 D n BKCMP>=(P) S1 S2 D n	• "n" points of data from S1, are compared with "n" points of data from S2 in 1 word units, and the comparison result is stored in the "n" points starting from the bit device designated by "D".

(2) Arithmetic operation instructions

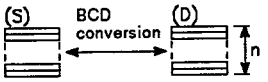
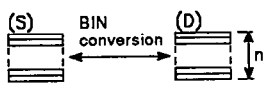
Classification	Symbol	Contents of Processing
BIN 16-bit addition /subtraction	+(P) S D	• (D) + (S) → (D)
	+(P) S1 S2 D	• (S1) + (S2) → (D)
	−(P) S D	• (D) − (S) → (D)
	−(P) S1 S2 D	• (S1) − (S2) → (D)

Classification	Symbol	Contents of Processing
BIN 32-bit addition /subtraction		• (D + 1, D) + (S + 1, S) → (D + 1, D)
		• (S1 + 1, S1) + (S2 + 1, S2) → (D + 1, D)
		• (D + 1, D) − (S + 1, S) → (D + 1, D)
		• (S1 + 1, S1) − (S2 + 1, S2) → (D + 1, D)
BIN 16-bit multiplication /division		• (S1) x (S2) → (D + 1, D)
		• (S1)/(S2) → quotient (D), remainder (D + 1)
BIN 32-bit multiplication /division		• (S1 + 1, S1) x (S2 + 1, S2) → (D + 3, D + 2, D + 1, D)
		• (S1 + 1, S1)/(S2 + 1, S2) → quotient (D + 1, D), remainder (D + 3, D + 2)
BCD 4-digit addition /subtraction		• (D) + (S) → (D)
		• (S1) + (S2) → (D)
		• (D) − (S) → (D)
		• (S1) − (S2) → (D)
BCD 8-digit addition /subtraction		• (D + 1, D) + (S + 1, S) → (D + 1, D)
		• (S1 + 1, S1) + (S2 + 1, S2) → (D + 1, D)
		• (D + 1, D) − (S + 1, S) → (D + 1, D)
		• (S1 + 1, S1) − (S2 + 1, S2) → (D + 1, D)
BCD 4-digit multiplication /division		• (S1) x (S2) → (D + 1, D)
		• (S1)/(S2) → quotient (D), remainder (D + 1)
BCD 8-digit multiplication /division		• (S1 + 1, S1) x (S2 + 1, S2) → (D + 3, D + 2, D + 1, D)
		• (S1 + 1, S1)/(S2 + 1, S2) → quotient (D + 1, D), remainder (D + 3, D + 2)
Floating decimal point data addition /subtraction		• (D + 1, D) + (S + 1, S) → (D + 1, D)
		• (S1 + 1, S1) + (S2 + 1, S2) → (D + 1, D)
		• (D + 1, D) − (S + 1, S) → (D + 1, D)
		• (S1 + 1, S1) − (S2 + 1, S2) → (D + 1, D)
Floating decimal point data multiplication /division		• (S1 + 1, S1) x (S2 + 1, S2) → (D + 1, D)
		• (S1 + 1, S1)/(S2 + 1, S2) → quotient (D + 1, D)
Character string data addition		• Joins the character string designated at (S) to the character string designated at (D) and stores the result from (D) onward.
		• Joins the character string designated at (S2) to the character string designated at (S1) and stores the result from (D) onward.

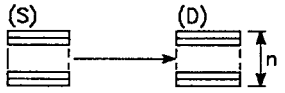
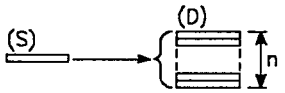
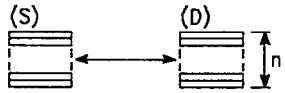
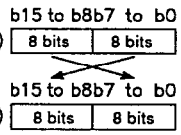
Classification	Symbol	Contents of Processing
BIN block addition /subtraction	BK+ S1 S2 D n	• Adds "n" points of data from (S1) and "n" points of data from (S2) in a batch and stores the result from (D) onward.
	BK– S1 S2 D n	
BIN data increment	INC(P) D	• (D) + 1 → (D)
	DINC(P) D	• (D + 1, D) + 1 → (D)
BIN data decrement	DEC(P) D	• (D) – 1 → (D)
	DDEC(P) D	• (D + 1, D) – 1 → (D)

(3) Data conversion instructions

Classification	Symbol	Contents of Processing
BCD conversion	BCD(P) S D	• (S) —BCD conversion→ (D) (S): BIN (0 to 9999)
	DBCD(P) S D	• (S + 1, S) —BCD conversion→ (D + 1, D) (S + 1, S): BIN (0 to 99999999)
BIN conversion	BIN(P) S D	• (S) —BIN conversion→ (D) (S): BCD (0 to 9999)
	DBIN(P) S D	• (S + 1, S) —BIN conversion→ (D + 1, D) (S + 1, S): BCD (0 to 99999999)
Floating decimal point → BIN conversion	INT(P) S D	• (S + 1, S) —BIN conversion→ (D) (S + 1, S): Real number (–32768 to 32767)
	DINT(P) S D	• (S + 1, S) —BIN conversion→ (D) (S + 1, S): Real number (–2147483648 to 2147483647)
BIN → floating decimal point conversion	FLT(P) S D	• (S + 1, S) —Floating decimal point conversion→ (D) (S + 1, S): Real number (–32768 to 32767)
	DFLT(P) S D	• (S + 1, S) —Floating decimal point conversion→ (D + 1, D) (S + 1, S): Real number (–2147483648 to 2147483647)

Classification	Symbol	Contents of Processing
BIN 16-bit ↔ 32-bit conversion	DBL(P) S D	• (S) —Conversion to 32-bit data→ (D + 1, D) (S): BIN (–32768 to 327767)
	WORD(P) S D	• (S + 1, S) —16-bit data conversion→ (D) (S + 1, S): BIN (–32768 to 32767)
BIN → gray code conversion	GRY(P) S D	• (S) —Gray code conversion→ (D) (S): BIN (–32768 to 327767)
	DGRY(P) S D	• (S + 1, S) —Gray code conversion→ (+1, DD) (S + 1, S): BIN (–32768 to 327767)
Gray code → BIN conversion	GBIN(P) S D	• (S) —Gray code conversion→ (D) (S): Gray code (–32768 to 327767)
	DGBIN(P) S D	• (S + 1, S) —Gray code conversion→ (D + 1, D) (S + 1, S): Gray code (–2147483648 to 2147483647)
2's complement	NEG(P) D	• $\overline{(D)}$ → (D) $\overline{(D)}$: BIN data
	DNEG(P) D	• $\overline{(D + 1, D)}$ → (D + 1, D) $\overline{(D + 1, D)}$: BIN data
	ENEG(P) D	• $\overline{(D + 1, D)}$ → (D + 1, D) $\overline{(D + 1, D)}$: Real number data
Block conversion	BKBCD(P) S D n	• Converts "n" points of BIN data from (S) in a batch to BCD data and stores the result from (D) onward. 
	BKBIN(P) S D n	• Converts "n" points of BCD data from (S) in a batch to BIN data and stores the result from (D) onward. 

(4) Data transfer instructions

Classification	Symbol	Contents of Processing
16-bit data transfer	MOV(P) S D	• (S) ⟶ (D)
32-bit data transfer	DMOV(P) S D	• (S + 1, S) ⟶ (D + 1, D)
Floating decimal point data transfer	EMOV(P) S D	• (S + 1, S) ⟶ (D + 1, D)
Character string data transfer	$MOV(P) S D	• Transfers the character string designated at "S" to devices starting with the device designated at "D".
16-bit data negation transfer	CML(P) S D	• $\overline{(S)}$ ⟶ (D)
32-bit data negation transfer	DCML(P) S D	• $\overline{(S + 1, S)}$ ⟶ (D + 1, D)
Data block transfer	BMOV(P) S D n	
Same data block transfer	FMOV(P) S D n	
16-bit data exchange	XCH(P) S D	• (S) ⟷ (D)
32-bit data exchange	DXCH(P) S D	• (S + 1, S) ⟷ (D + 1, D)
Block data exchange	BXCH(P) S D n	
Vertical byte exchange	SWAP(P) S D	

(5) Program branch instructions

Classification	Symbol	Contents of Processing
Jump	CJ Pn	• Jumps to Pn on satisfaction of the input condition.
	SCJ Pn	• Jumps to Pn beginning with the scan after the one in which the input condition is satisfied.
	JMP Pn	• Unconditionally jumps to Pn.
	GOEND	• Jumps to the END instruction on satisfaction of the input condition.

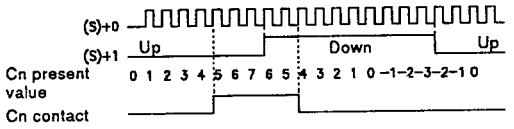
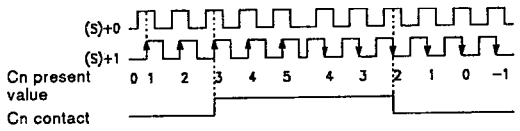
(6) Program execution control instructions

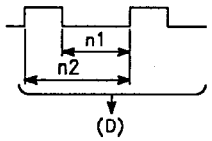
Classification	Symbol	Contents of Processing
Disable interrupt	DI	• Disables execution of interrupt programs.
Enable interrupt	EI	• Cancels the execution disabled status for interrupt programs.
Interrupt disable /enable setting	IMASK S	• Disables or enables execution of individual interrupt programs.
Return	IRET	• Returns execution from the interrupt program to the sequence program.

(7) I/O refresh instructions

Classification	Symbol	Contents of Processing
I/O refresh	RFS D n	• Executes partial refresh for the designated I/O part way through a scan.

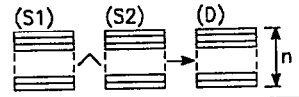
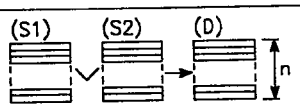
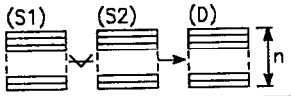
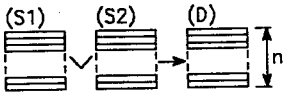
(8) Other convenient instructions

Classification	Symbol	Contents of Processing
Up/down counter	UDCNT1 S D n	
	UDCNT2 S D n	
Teaching timer	TTMR D n	• (TTMR ON time) x n → (D) ↑ n=0:1, n=0:10, n=2:100
Special timer	STMR S n D	• The four bits starting with the bit designated at "D" perform the following operations in accordance with the ON/OFF status of the STMR instruction. (D) + 0: Off delay timer output (D) + 1: Once-only timer output after turning OFF (D) + 2: Once-only timer output after turning ON (D) + 3: ON delay timer
Shortest path control	ROTC S n1 n2 D	• Rotates a rotary table that indexes in increments of "n1" from the position at which it is stopped to the position designated by "S+1" in the direction that gives the shortest travel.
Ramp signal	RAMP n1 n2 D1 n3 D2	• Changes the device data designated at D1 in the range n1 to n2 in n3 scans.
Pulse density	SPD S n D	• Counts the pulse input of the device designated at "S" at the time designated at "n" and stores the result in the device designated at "D".
Pulse output	PLSY n1 n2 D	• (n1)Hz ——→ (D) Output "n2" times.

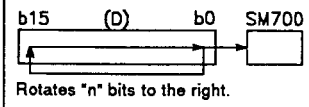
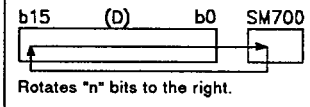
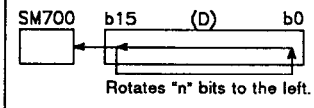
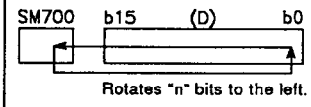
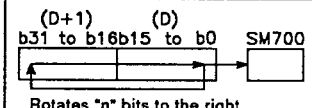
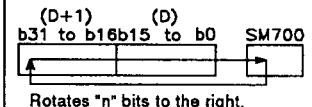
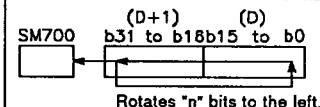
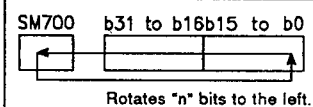
Classification	Symbol	Contents of Processing
Pulse width modulation	PWM n1 n2 D	
Matrix input	MTR S D1 D2 n	• Consecutively inputs the data of n rows of 16 devices starting from the device designated at "S" and stores it in devices starting from the device designated at D2.

1.3 Application Instructions

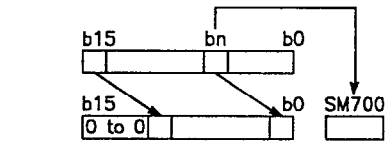
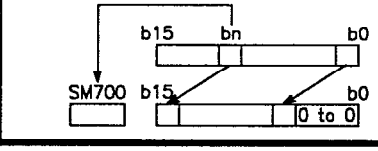
(1) Logical operation instructions

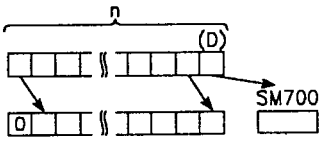
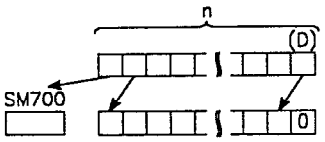
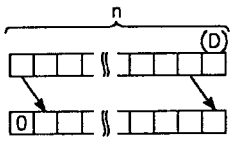
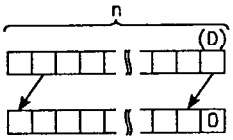
Classification	Symbol	Contents of Processing
Logical product	WAND(P) S D	• (D) ∧ (S) → (D)
	WAND(P) S1 S2 D	• (S1) ∧ (S2) → (D)
	DAND(P) S D	• (D + 1, D) ∧ (S + 1, S) → (D + 1, D)
	DAND(P) S1 S2 D	• (S1 + 1, S1) ∧ (S2 + 1, S2) → (D + 1, D)
	BKAND(P) S1 S2 D n	
Logical sum	WOR(P) S D	• (D) ∨ (S) → (D)
	WOR(P) S1 S2 D	• (S1) ∨ (S2) → (D)
	DOR(P) S D	• (D + 1, D) ∨ (S + 1, S) → (D + 1, D)
	DOR(P) S1 S2 D	• (S1 + 1, S1) ∨ (S2 + 1, S2) → (D + 1, D)
	BKOR(P) S1 S2 D n	
Exclusive logical sum	WXOR(P) S D	• (D) ⊻ (S) → (D)
	WXOR(P) S1 S2 D	• (S1) ⊻ (S2) → (D)
	DXOR(P) S D	• (D + 1, D) ⊻ (S + 1, S) → (D + 1, D)
	DXOR(P) S1 S2 D	• (S1 + 1, S1) ⊻ (S2 + 1, S2) → (D + 1, D)
	BKXOR(P) S1 S2 D n	
NOT exclusive logical sum	WNXR(P) S D	• $\overline{(D) \veebar (S)}$ → (D)
	WNXR(P) S1 S2 D	• $\overline{(S1) \veebar (S2)}$ → (D)
	DNXR(P) S D	• $\overline{(D + 1, D) \veebar (S + 1, S)}$ → (D + 1, D)
	DNXR(P) S1 S2 D	• $\overline{(S1 + 1, S1) \veebar (S2 + 1, S2)}$ → (D + 1, D)
	BKNXR(P) S1 S2 D n	

(2) Rotation instructions

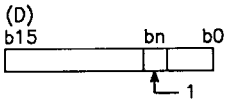
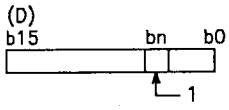
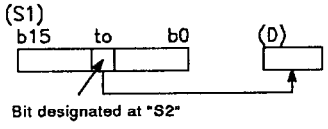
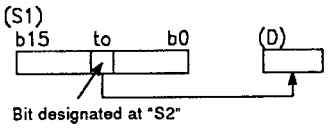
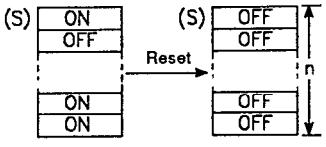
Classification	Symbol	Contents of Processing
Right rotation	ROR(P) D n	
	RCR(P) D n	
Left rotation	ROL(P) D n	
	RCL(P) D n	
Right rotation	DROR(P) D n	
	DRCR(P) D n	
Left rotation	DROL(P) D n	
	DRCL(P) D n	

(3) Shift instructions

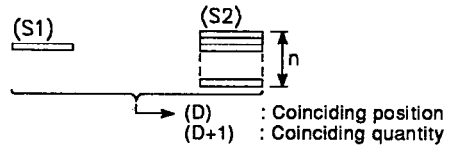
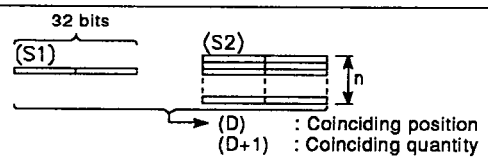
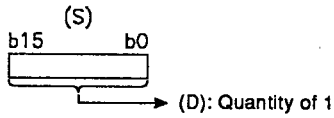
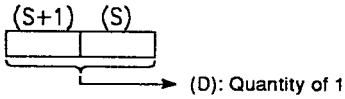
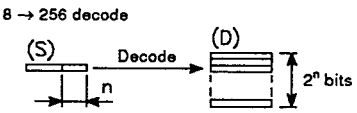
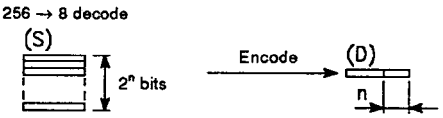
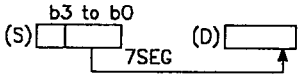
Classification	Symbol	Contents of Processing
n bit shift	SFR(P) D n	
	SFL(P) D n	

Classification	Symbol	Contents of Processing
1 bit shift	BSFR(P) D n	
	BSFL(P) D n	
1 word shift	DSFR(P) D n	
	DSFL(P) D n	

(4) Bit processing instructions

Classification	Symbol	Contents of Processing
Bit set/reset	BSET(P) D n	
	BRST(P) D n	
Bit test	TEST(P) S1 S2 D	
	DTEST(P) S1 S2 D	
Bit device batch reset	BKRST(P) S n	

(5) Data processing instructions

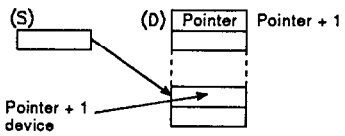
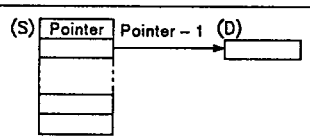
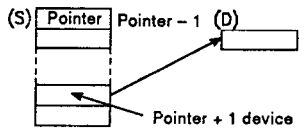
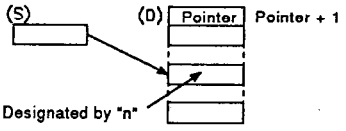
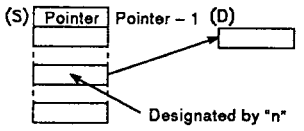
Classification	Symbol	Contents of Processing
Data search	SER(P) S1 S2 D n	
	DSER(P) S1 S2 D n	
Bit check	SUM(P) S D	
	DSUM(P) S D	
Decode	DECO(P) S D n	
Encode	ENCO(P) S D n	
7 segment decode	SEG(P) S D	
Dissociation /Association	DIS(P) S D n	• Dissociates the 16-bit data designated at "S" into 4-bit units, and stores these data in the least significant four bits of n devices starting with the one designated at "D". (n ≤ 4)
	UNI(P) S D n	• Associates the least significant 4-bit data of n devices starting from the device designated at "S" by storing this data in the device designated at "D". (n ≤ 4)
	NDIS(P) S1 D S2	• Dissociates the data of the devices starting with the device designated at "S1" into the designated bits starting with the device designated by "S2", and stores this data in sequence starting at the device designated at "D".
	NUNI(P) S1 D S2	• Associates each of the bits designated from "S2" onward in the data of the devices starting from the device designated at "S1", by storing this data in sequence starting at the device designated at "D".
	WTOB(P) S D n	• Dissociates the 16-bit data that starts from the device designated at "S" into 8-bit units, and stores data corresponding to "n" points in sequence starting from the device designated at "D".
	BTOW(P) S D n	• Associates the least significant 8 bits of "n" points of 16-bit data starting from the device designated at "S" to give 16-bit data, and stores this in sequence starting from the device designated at "D".

Classification	Symbol	Contents of Processing
Search	MAX(P) S D n	• Searches the "n" points of data starting from the device designated at "S" in 16-bit units, and stores the maximum value in the device designated at "D".
	MIN(P) S D n	• Searches the "n" points of data starting from the device designated at "S" in 16-bit units, and stores the minimum value in the device designated at "D".
	DMAX(P) S D n	• Searches the "2 x n" points of data starting from the device designated at "S" in 32-bit units, and stores the maximum value in the device designated at "D".
	DMIN(P) S D n	• Searches the "2 x n" points of data starting from the device designated at "S" in 32-bit units, and stores the minimum value in the device designated at "D".
Sort	SORT S1 n S2 D1 D2 •S2: Number of comparisons executed at one time •D1: Device turned ON on completion of sorting •D2: For system use	• Sorts "n" points of data starting from the device designated at "S1" in 16-bit units. [Max. number of scans required: {n x (n – 1)}/2 scans]
	DSORT S1 n S2 D1 D2 •S2: Number of comparisons executed at one time •D1: Device turned ON on completion of sorting •D2: For system use	• Sorts "2 x n" points of data starting from the device designated at "S1" in 32-bit units. [Max. number of scans required: {n x (n – 1)}/2 scans]

(6) Structured program instructions

Classification	Symbol	Contents of Processing
Repetition	FOR n NEXT	• Executes the program section between FOR and NEXT n times.
	BREAK(P) D Pn	• Forcibly ends execution of the program section between FOR and NEXT and causes a jump to Pn.
Subroutine program call	CALL(P) Pn S1 to Sn	• Executes the subroutine Pn on satisfaction of the input condition. (S1 to Sn is the argument for the subroutine program. $0 \leq n \leq 5$)
	RET	• Causes a return from the subroutine program.
	FCALL(P) Pn S1 to Sn	• Executes non-execution processing for subroutine program Pn when the input condition is not satisfied.
	ECALL(P) * Pn S1 to Sn *: Program name	• Executes the subroutine Pn in the designated program on satisfaction of the input condition. (S1 to Sn is the argument for the subroutine program. $0 \leq n \leq 5$)
	EFCALL(P) * Pn S1 to Sn *: Program name	• Executes non-execution processing for the subroutine Pn in the designated program when the input condition is not satisfied.
	COM	• Executes link refresh and general data processing.
Ladder index qualification	IX S Device qualification ladder IXEND	• Executes index qualification for each of the devices used in the device qualification ladder.
Ladder index qualification	IXDEV IXSET Pn D Designation of qualification value	• Stores the qualification value for index qualification at IX to IXEND in devices starting with the device designated at "D".

(7) Table operation instructions

Classification	Symbol	Contents of Processing
Table processing	FIFW(P) S D	
	FIFR(P) S D	
	FPOR(P) S D	
	FINS(P) S D n	
	FDEL(P) S D n	

(8) Buffer memory access instructions

Classification	Symbol	Contents of Processing
Data read	FROM n1 n2 D n3	• Reads data in 16-bit units from special function modules.
	DFRO n1 n2 D n3	• Reads data in 32-bit units from special function modules.
Data write	TO(P) n1 n2 S n3	• Writes data in 16-bit units to special function modules.
	DTO(P) n1 n2 S n3	• Writes data in 32-bit units to special function modules.

(9) Display instructions

Classification	Symbol	Contents of Processing
ASCII print	*When SM701 is OFF PR S D	• Outputs 8 points (16 characters) of ASCII code, starting from the device designated at "S", to an output module.
	*When SM701 is ON PR S D	• Outputs the ASCII code in devices starting from the device designated at "S" and ending at 00_H, to an output module.
	PRC S D	• Converts the device comment designated at "S" to ASCII code and outputs the result to an output module.
Display	LED S	• Displays the ASCII code in 8 points (corresponding to 16 characters), starting from the device designated at "S", on the LED display.
	LEDC S	• Displays the comment of the device designated at "S" on the LED display.
Reset	LEDR	• Resets annunciators and resets the LED display.

(10) Debugging and fault diagnosis instructions

Classification	Symbol	Contents of Processing
Fault check	CHKST	• When the CHKST instruction is executed, the CHK instruction is executed. • When the CHKST instruction is not executed, operation jumps to the step following the step containing the CHK instruction.
	CHK (Check condition)	• When normal → SM80: OFF, SD80: 0 • When abnormal → SM80: ON, SD80: fault No.
	CHKCIR	• Start of the ladder pattern change to be checked in accordance with the CHK instruction.
	CHKEND	• End of the ladder pattern change to be checked in accordance with the CHK instruction.
Status latch	SLT	• Executes status latch.
	SLTR	• Resets the status latch and enables re-execution of status latch.
Sampling trace	STRA	• Triggers sampling trace.
	STRAR	• Resets the sampling trace and enables re-execution of sampling trace.
Program trace	PTRA	• Triggers program trace.
	PTRAR	• Resets the program trace and enables re-execution of program trace.
	PTRAEXE	• Executes program trace.

(11) Character string processing instructions

Classification	Symbol	Contents of Processing
BIN ↓ Decimal ASCII	BINDA(P) S D	• Converts the 1 word of BIN data designated at "S" into a 5-digit decimal ASCII value, and stores this from the word device number designated at "D".
	DBINDA(P) S D	• Converts the 2 words of BIN data designated at "S" into a 10-digit decimal ASCII value, and stores this from the word device number designated at "D".
BIN ↓ Hexadecimal ASCII	BINHA(P) S D	• Converts the 1 word of BIN data designated at "S" into a 4-digit hexadecimal ASCII value, and stores this from the word device number designated at "D".
	DBINHA(P) S D	• Converts the 2 words of BIN data designated at "S" into an 8-digit hexadecimal ASCII value, and stores this from the word device number designated at "D".
BCD ↓ ASCII	BCDDA(P) S D	• Converts the 1-word BCD value designated at "S" into a 4-digit decimal ASCII value, and stores this from the word device number designated at "D".
	DBCDDA(P) S D	• Converts the 2-word BCD value designated at "S" into an 8-digit decimal ASCII value, and stores this from the word device number designated at "D".
Decimal ASCII ↓ BIN	DABIN(P) S D	• Converts the 5-digit decimal ASCII value designated at "S" to a 1-word BIN value, and stores this at the word device number designated at "D".
	DDABIN(P) S D	• Converts the 10-digit decimal ASCII value designated at "S" to a 2-word BIN value, and stores this at the word device number designated at "D".

Classification	Symbol	Contents of Processing
Hexadecimal ASCII ↓ BIN	HABIN(P) S D	• Converts the 4-digit hexadecimal ASCII value designated at "S" to a 1-word BIN value, and stores this at the word device number designated at "D".
	DHABIN(P) S D	• Converts the 8-digit decimal ASCII value designated at "S" to a 2-word BIN value, and stores this at the word device number designated at "D".
ASCII ↓ BCD	DABCD(P) S D	• Converts the 4-digit decimal ASCII value designated at "S" to a 1-word BCD value, and stores this at the word device number designated at "D".
	DDABCD(P) S D	• Converts the 8-digit decimal ASCII value designated at "S" to a 2-word BCD value, and stores this at the word device number designated at "D".
Device comment read	COMRD(P) S D	• Stores the comment data of the device designated at "S" to the device designated at "D".
Character string length detection	LEN(P) S D	• Stores the length of the character string data (number of characters) that is stored in the device designated at "S" in the device designated at "D".
BIN ↓ Decimal character string	STR(P) S1 S2 D	• Converts the 1-word BIN value designated at "S2" into a decimal character string with the total number of digits and number of fraction part digits designated at "S1", and stores it in the device designated at "D".
	DSTR(P) S1 S2 D	• Converts the 2-word BIN value designated at "S2" into a decimal character string with the total number of digits and number of fraction part digits designated at "S1", and stores it in the device designated at "D".
Decimal character string ↓ BIN	VAL(P) S D1 D2	• Converts the character string that contains a decimal point designated at "S" to a 1-word BIN value and number of fraction part digits, and stores them in the devices designated at "D1" and "D2".
	DVAL(P) S D1 D2	• Converts the character string that contains a decimal point designated at "S" to a 2-word BIN value and number of fraction part digits, and stores them in the devices designated at "D1" and "D2".
Floating decimal point ↓ Character string	ESTR(P) S1 S2 D	• Converts the floating decimal point data designated at "S" to a character string and stores it in the device designated at "D".
Character string ↓ Floating decimal point	EVAL(P) S D	• Converts the character string designated at "S" to floating decimal point data and stores it in the device designated at "D".
Hexadecimal BIN ↓ ASCII	ASC(P) S D n	• Converts the 1-word BIN value starting from the device number designated at "S" to hexadecimal ASCII, and stores it in the number of characters designated at "n" from the word device number designated at "D".
ASCII ↓ Hexadecimal BIN	HEX(P) S D n	• Converts the number of characters designated at "n" of the hexadecimal ASCII data starting from the word device designated at "S", to a BIN value, and stores it from the device number designated at "D" onward.

Classification	Symbol	Contents of Processing
Character string processing	RIGHT(P) S D n	• Stores "n" characters from the final character of the word device designated at "S" to the device designated at "D".
	LEFT(P) S D n	• Stores "n" characters from the initial character of the word device designated at "S" to the device designated at "D".
	MIDR(P) S1 D S2	• Stores the designated number of characters from the position designated at "S2" of the character string designated at "S1" to the device designated at "D".
	MIDW(P) S1 D S2	• Stores the character string designated at "S1" to the designated number of characters from the position designated at "S2" of the device designated at "D".
	INSTR(P) S1 S2 D n	• Searches for character string S1 starting from the nth character of character string S2 and stores the position where a match is found to "D".
Floating decimal point ↓ BCD resolution	EMOD(P) S1 S2 D	• Converts the floating decimal point data designated at "S1" to BCD data with the number of fraction part digits designated at "S2", and stores this data in the device designated at "D".
BCD ↓ floating decimal point	EREXP(P) S1 S2 D	• Converts the BCD data of "S1" to floating decimal point data with the number of fraction part digits designated at "S2" and stores this data in the device designated at "D".

(12) Special function instructions

Classification	Symbol	Contents of Processing
Trigonometric functions (floating decimal point data)	SIN(P) S D	• Sin(S + 1, S) → (D + 1, D)
	COS(P) S D	• Cos(S + 1, S) → (D + 1, D)
	TAN(P) S D	• Tan(S + 1, S) → (D + 1, D)
	ASIN(P) S D	• Sin^{-1}(S + 1, S) → (D + 1, D)
	ACOS(P) S D	• Cos^{-1}(S + 1, S) → (D + 1, D)
	ATAN(P) S D	• Tan^{-1}(S + 1, S) → (D + 1, D)
Degree ↔ radian conversion	RAD(P) S D	• (S + 1, S) → (D + 1, D) Degree → radian conversion
	DEG(P) S D	• (S + 1, S) → (D + 1, D) Radian → degree conversion
	SQR(P) S D	• $\sqrt{(S + 1, S)}$ → (D + 1, D)
Exponent operation	EXP(P) S D	• $e^{(S + 1, S)}$ → (D + 1, D)
Natural logarithm	LOG(P) S D	• Log e(S + 1, S) → (D + 1, D)
Square root	BSQR(P) S D	• $\sqrt{(S)}$ → (D) + 0 Integer part + 1 Fraction part
	BDSQR(P) S D	• $\sqrt{(S + 1, S)}$ → (D) + 0 Integer part + 1 Fraction part

Classification	Symbol	Contents of Processing
Trigonometric functions	BSIN(P) S D	• Sin (S) → (D) + 0 Sign + 1 Integer part + 2 Fraction part
	BCOS(P) S D	• Cos (S) → (D) + 0 Sign + 1 Integer part + 2 Fraction part
	BTAN(P) S D	• Tan (S) → (D) + 0 Sign + 1 Integer part + 2 Fraction part
	BASIN(P) S D	• Sin^{-1} (S) → (D) + 0 Sign + 1 Integer part + 2 Fraction part
	BACOS(P) S D	• Cos^{-1} (S) → (D) + 0 Sign + 1 Integer part + 2 Fraction part
	BATAN(P) S D	• Tan^{-1} (S) → (D) + 0 Sign + 1 Integer part + 2 Fraction part

(13) Data control instructions

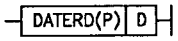
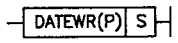
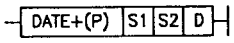
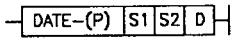
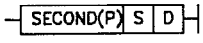
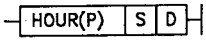
Classification	Symbol	Contents of Processing
Upper/lower limit control	LIMIT(P) S1 S2 S3 D	• Processes the value designated at "S3" to comply with a fixed range of data defined by the upper and lower limits set at "S1" and "S2", and stores the result at the word device number designated at "D". • When S3 < S1 ····· The value at "S1" is stored in "D". • When S1 ≤ S3 ≤ S2 ····· The value at "S3" is stored in "D". • When S2 < S3 ····· The value at "S2" is stored in "D".
	DLIMIT(P) S1 S2 S3 D	• Processes the value designated at (S3 + 1, S3) to comply with a fixed range of data defined by the upper and lower limits set at (S1 + 1, S1) and (S2 + 1, S2), and stores the result at the word device designated at (D + 1, D). • When (S3 + 1, S3) < (S1 + 1, S1) ····· The (S1 + 1, S1) value is stored in (D + 1, D). • When (S1 + 1, S1) ≤ (S3 + 1, S3) ≤ (S2 + 1, S2) ····· The (S3 + 1, S3) value is stored in (D + 1, D). • When (S2, S2 + 1) < (S3, S3 + 1) ····· The (S2 + 1, S2) value is stored in (D + 1, D).
Dead band control	BAND(P) S1 S2 S3 D	• Taking the area set by S1 and S2 as the dead band, if the input value designated at "S3" is within the dead band, "0" is stored at the word device number designated at "D" and if it is outside the dead band, the value obtained by subtracting the dead band upper/lower limit value from the input value is stored at the word device number designated at "D". • When S1 ≤ S3 ≤ S2 ··· 0 → D • When S3 < S1 ····· S3 − S1 → D • When S3 > S2 ····· S3 − S2 → D
	DBAND(P) S1 S2 S3 D	• Taking the area set by (S1 + 1, S1) and (S2 + 1, S2) as the dead band, if the input value designated at (S3 + 1, S3) is within the dead band, "0" is stored at the word device number designated at "D" and if it is outside the dead band, the value obtained by subtracting the dead band upper/lower limit value from the input value is stored to the word device whose number is designated at "D". • When (S1 + 1, S1) ≤ (S3 + 1, S3) ≤ (S2 + 1, S2) ····· 0 → (D+1, D) • When (S3 + 1, S3) < (S1 + 1, S1) ····· (S3 + 1, S3) − (S1 + 1, S1) → (D + 1, D) • When (S3 + 1, S3) > (S2 + 1, S2) ····· (S3 + 1, S3) − (S2 + 1, S2) → (D + 1, D)

Classification	Symbol	Contents of Processing
Zone control	ZONE(P) S1 S2 S3 D	• By setting positive and negative bias values for the input value designated at "S3" in "S1" and "S2", calculates the value for S1 + bias, and stores it to the word device whose number is designated at "D". • When S3 = 0 ····0 → D • When S3 > 0 ··· S3 + S2 → D • When S3 < 0 ····· S3 – S1 → D
	DZONE(P) S1 S2 S3 D	• By setting positive and negative bias values for the input value designated at (S3 + 1, S3) in (S1 + 1, S1) and (S2 + 1, S2), calculates the value for S1 + bias, and stores it to the word device whose number is designated at (D + 1, D). • When (S3 + 1, S3) = 0 ··· 0 → (D + 1, D) • When (S3 + 1, S3) > 0 ····· (S3 + 1, S3) – (S2 + 1, S2) → (D + 1, D) • When (S3 + 1, S3) < 0 ····· (S3 + 1, S3) + (S1 + 1, S1) → (D + 1, D)

(14) Switching instructions

Classification	Symbol	Contents of Processing
Block No. setting	RSET(P) S	• Changes the block No. of an extension file register to the number designated at "S".
	QDRSET(P) File name	• Sets the name of a file to be used as a file register.
	QCDSET(P) File name	• Sets the name of a file to be used as a comment register.

(15) Clock instructions

<table>
<tr><th>Classification</th><th>Symbol</th><th>Contents of Processing</th></tr>
<tr><td rowspan="2">Clock data read/write</td><td> 
</td><td>• (Clock device) → (D) + 0 Year
+ 1 Month
+ 2 Day of month
+ 3 Hour
+ 4 Minute
+ 5 Second
+ 6 Day of week</td></tr>
<tr><td> 
</td><td>• (S) + 0 Year
+ 1 Month
+ 2 Day of month
+ 3 Hour
+ 4 Minute
+ 5 Second
+ 6 Day of week
→ (Clock device)</td></tr>
<tr><td rowspan="4">Clock data read/write</td><td> 
</td><td>(S1) Hour, Minute, Second + (S2) Hour, Minute, Second → (D) Hour, Minute, Second</td></tr>
<tr><td> 
</td><td>(S1) Hour, Minute, Second − (S2) Hour, Minute, Second → (D) Hour, Minute, Second</td></tr>
<tr><td> 
</td><td>(S) Hour, Minute, Second → (D) Second (lower level), Second (Upper level)</td></tr>
<tr><td> 
</td><td>(S) Second (lower level), Second (Upper level) → (D) Hour, Minute, Second</td></tr>
</table>

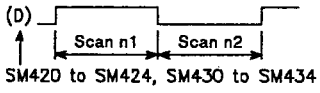
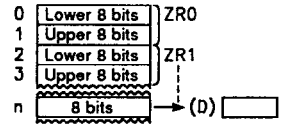
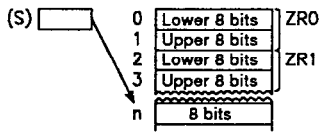
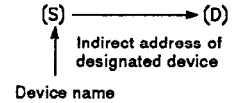
(16) Instructions for peripheral devices

Classification	Symbol	Contents of Processing
Input/output to peripheral device	MSG S	• Stores the message designated at S to the QnACPU. This message is displayed at the peripheral device.
	PKEY D	• Stores the data input from a peripheral device to the device designated at "D".

(17) Program instructions

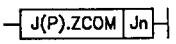
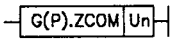
Classification	Symbol	Contents of Processing
Program execution status switch	PSTOP(P) Program name	• Sets the designated program in the standby status.
	POFF(P) Program name	• Turns OFF the coil of the designated program's OUT instruction and sets the program to the standby status.
	PSCAN(P) Program name	• Registers the designated program as a scan execution type program.
	PLOW(P) Program name	• Registers the designated program as a low-speed execution type program.

(18) Other instructions

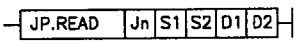
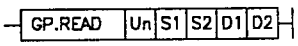
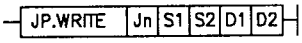
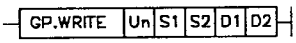
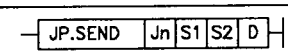
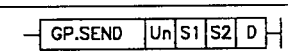
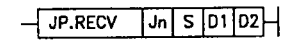
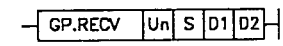
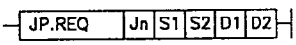
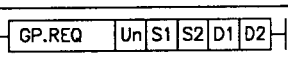
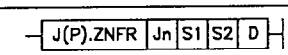
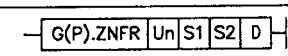
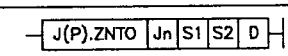
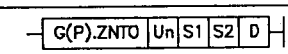
Classification	Symbol	Contents of Processing
WDT reset	WDT(P)	• Sets the WDT in a sequence program.
Timing clock	DUTY n1 n2 D	
Direct read/write in byte units	ZRRDB(P) n D	
Direct read/write in 1 byte units	ZRWRB(P) n S	
Indirect address set	ADRSET(P) S D	
Numeric key input from keyboard	KEY S n D1 D2	• Fetches ASCII data of the input module designated at "S" in 8-point units, converts this data to hexadecimal values and stores them in devices starting with the device whose number is designated at "D1".

1.4 Data Link Instructions

(1) Link refresh instructions

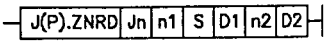
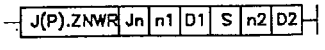
Classification	Symbol	Contents of Processing
Refresh designated network		• Refreshes the network module corresponding to the designated network No. in network n.
		• Refreshes the network module corresponding to the designated I/O number in network n.

(2) QnA link dedicated instructions

Classification	Symbol	Contents of Processing
Other station data read/write		• Reads data from word devices at another station.
		
		• Writes data to word devices at another station.
		
Sending/receiving data to/from other stations		• Sends data (message) to another station.
		
		• Receives data (message) from another station.
		
Processing request to other station		• Executes remote RUN/STOP with respect to another station.
		
Data read/write from a special function module at a remote I/O station		• Reads data from a special function module installed at a remote station in a MELSECNET/10 network.
		
		• Writes data to a special function module at a remote station in a MELSECNET/10 network.
		

(GP. *** instructions can also be used for AJ71QC24)

(3) A series link instructions

Classification	Symbol	Contents of Processing
Read word device of designated station		• Reads the data of T, C, D, and W devices of other stations in a MELSECNET(II) or MELSECNET/10 system.
Write word device to designated station		• Reads the data of T, C, D, and W devices of other stations.

Classification	Symbol	Contents of Processing
Read/write data from/to special function module in remote I/O station	G(P).RFRP Un n1 D1 n2 D2	• Reads data from a special function module installed at a remote I/O station in a MELSECNET(II) system
	G(P).RTOP Un n1 S n2 D	• Writes data to a special function module installed at a remote I/O station in a MELSECNET(II) system.

(4) Routing parameter instructions

Classification	Symbol	Contents of Processing
Read routing information	Z.RTREAD n D	• Reads the data of the transfer destination network number designated at "n" from the routing parameters and stores it from "D" onward.
Register routing information	Z.RTWRITE n S	• Registers the routing parameter data from S onward in the area for the transfer destination network number designated at "n" , in the parameters.

1.5 Instructions for PID Control

Classification	Symbol	Contents of Processing
Set data for PID control	PIDINIT S	Registers the PID control data in devices from the one whose number is set at "S" onward in the PC CPU.
Execute PID control	PIDCONT S	Performs PID operation on the basis of the set value (SV) and process value (PV) set from the device whose number is designated at "S" onward, and stores the operation result in the manipulated value (MV) area.
PID control status monitor	PID57 n S1 S2	Displays, in the form of a bar graph, the PID control status for the loop No. designated at "S1" on the display of the AD57 designated at "n". At the start of execution of PID control monitor, static image elements apart from the bar graph and numerical data are displayed by issuing the initial screen display request designated at "S2".
Stop operation of designated loop	PIDSTOP n	Stops operation of the loop whose number is designated at "n".
Start operation of designated loop	PIDRUN n	Starts operation of the loop whose number is designated at "n".
Designated loop parameter change	PIDPRMW n S	Changes the operation parameters of the loop whose number is designated at "n" to the data set in the devices starting from the one whose number is designated at "S".

1.6 Special Function Module Instructions

Classification	Function	Instruction Symbol
Control instructions for AD61(S1)	Setting preset data	RVWR1, PVWR2
	Setting the set value data for larger/ smaller/matched judgments	SVWR1, SVWR2
	Reading present values	PVRD1, PVRD2
Control instructions for AD59(S1)	Outputting required number of characters to a printer	PRN
	Outputting characters up to the 00_H code to a printer	PR
	Reading data from a memory card	GET
	Writing data to a memory card	PUT
Control instructions for AJ71C24 (-S3/S6/S8)	Sending a designated number of bytes of data in the no-protocol mode	PRN
	Sending data up to the 00_H code in the no-protocol mode	PR
	Receiving data in the no-protocol mode	INPUT
	Reading the communications status	SPBUSY
	Forcibly suspending send/receive processing	SPCLR
Control instructions for AJ71C21(S1)	Sending the designated number of bytes of data	PRN2, PRN4
	Sending data up to the 00_H code	PR2, PR4
	Data receive	INPUT2, INPUT4
	Reading the RAM	GET
	Writing to the RAM	PUT
	Forcibly suspending communications processing	SPBUSY
Control instructions for AJ71PT32-S3	Key input from an operating box	INPUT
	Sending the designated number of bytes of data in the no-protocol mode	PRN
	Sending data up to the 00_H code in the no-protocol mode	PR
	Receiving data in the no-protocol mode	INPUT
	Communications with remote terminal modules	MINI, MINIEND
	Error reset with respect to remote terminal modules	MINIERR
	Reading the communications status	SPBUSY
	Forcibly suspending communications processing	SPCLR

Classification	Function	Instruction Symbol
Control instructions for AD57	Setting the display mode	CMODE
	Displaying a canvas screen	CPS1
	Changing the VRAM display address	CPS2
	Transferring canvas data to the VRAM area	CMOV
	Clearing the display area	CLS
	Clearing the VRAM area	CLV
	Scrolling the screen	CSCRU, CSCRD
	Displaying the cursor	CON1, CON2
	Deleting the cursor	COFF
	Setting the cursor position	LOCATE
	Setting normal or highlighted display for characters	CNOR, CREV
	Switching between normal and highlighted display for characters	CRDSP, SRDSPV
	Designating the character display color	COLOR
	Changing the character color	CCDSP, CCDSPV
	Displaying ASCII characters	PR, PRN
	Writing ASCII characters to the VRAM	PRV, PRNV
	Displaying characters	EPR, EPRN
	Writing characters to the VRAM	EPRV, EPRNV
	Consecutive display of the same character	CR1, CR2, CC1, CC2
	Displaying "–" (minus)	CINMP
	Displaying "-" (hyphen)	CINHP
	Displaying "." (period, decimal point)	CINPT
	Displaying numerals	CIN0 to CIN9
	Displaying letters of the alphabet	CINA to CINZ
	Displaying spaces	CINSP
	Clearing display of designated area	CINCLR
	ASCII code conversion of designated character strings	INPUT
	Reading VRAM data	GET
	Writing VRAM data	PUT
	Reading the display status	STAT
Instructions for AJ71ID□-R4	ID controller initial setting	IDINIT1, IDINIT2
	Reading from the ID data carrier	IDRD1, IDRD2
	Writing to the ID data carrier	IDWD1, IDWD2
	Continuous reading from the ID data carrier	IDARD1, IDARD2
	Continuous writing to the ID data carrier	IDAWD1, IDAWD2
	Comparing data with the ID data carrier	IDCMP1, IDCMP2
	Batch writing the same data to the ID data carrier	IDFILL1, IDFILL2
	Copying between ID data carriers	IDCOPY1, IDCOPY2
	Clearing the ID data carrier	IDCLR1, IDCLR2
	Ending use of the ID data carrier	IDOFF1, IDOFF2
	Starting use of the ID data carrier	IDON1, IDON2

Classification	Function	Instruction Symbol
Instructions for AJ71QC24*	Writing a user-registered frame to the AJ71QC24's EEPROM	PUTE
	Reading the user-registered frame from the AJ71QC24's EEPROM	GETE
	Sending data in accordance with the dedicated protocol using the "on demand" function	ONDEMAND
	Sending the designated number of bytes of data in the no-protocol mode	OUTPUT
	Sending data in accordance with the send schedule table in the no-protocol mode	PRR
	Receiving data in the no-protocol mode	INPUT
	Sending data with the bi-directional protocol	BIDOUT
	Receiving data with the bi-directional protocol	BIDIN
	Reading the communications status	SPBUSY
	Reading devices from other stations	READ
	Writing devices to other stations	SWRITE
	Sending data to other stations	SEND
	Receiving data from other stations	RECV
	Sending transient transmission request data to other stations	REQ

* An AJ71QC24 can be used with QnA link instructions designated for use with special function modules (G(P). ***).

APPENDIX 2 SPECIAL RELAY LIST

Special relays, SM, are internal relays whose applications are fixed in the programmable controller.
For this reason, they cannot be used by sequence programs in the same way as the normal internal relays.
However, they can be turned ON or OFF as needed in order to control the QnACPU.

The headings in the table that follows have the following meanings.

Item	Function of Item
Number	• Indicates the number of the special relay.
Name	• Indicates the name of the special relay.
Meaning	• Indicates the nature of the special relay.
Explanation	• Contains detailed information about the nature of the special relay.
Set by (When set)	• Indicates whether the relay is set by the system or user, and, if it is set by the system, when setting is performed. <Set by> S : Set by system U : Set by user (in sequence program or test operation at a peripheral device) S/U : Set by both system and user <When set> → indicated only if setting is done by system. Each END : Set during each END processing Initial : Set only during initial processing (when power supply is turned ON, or when going from STOP to RUN) Status change : Set only when there is a change in status Error : Set when error is generated Instruction execution : Set when instruction is executed Request : Set only when there is a user request (through SM, etc.)
Corresponding ACPU M9□□□	• Indicates special relay M9□□□/corresponding to the ACPU. (Change and notation when there has been a change in contents) • Items indicated as "New" have been newly added for QnACPU

For details on the following items, see these manuals:
• Networks → MELSECNET/10 Network System Reference Manual for QnA
• SFC → QnACPU Programming Manual (SFC)

Special Relay List

(1) Diagnostic Information

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU M9□□□
SM0	Diagnostic errors	OFF : No error ON : Error	• ON if diagnosis results show error occurrence (Includes external diagnosis) • Stays ON subsequently even if normal operations restored	S (Error)	New
SM1	Self-diagnostic error	OFF : No self-diagnosis errors ON : Self-diagnosis	• Comes ON when an error occurs as a result of self -diagnosis. • Stays ON subsequently even if normal operations restored	S (Error)	M9008
SM5	Error common information	OFF : No error common information ON : Error common information	• When SM0 is ON, ON if there is error common information	S (Error)	New
SM16	Error individual information	OFF : No error individual information ON : Error individual information	• When SM0 is ON, ON if there is error individual information	S (Error)	New
SM50	Error reset	OFF → ON: Error reset	• Conducts error reset operation • See Chapter 5 for further information	U	New
SM51	Battery low latch	OFF : Normal ON : Battery low	• ON if battery voltage at CPU or memory card drops below rated value. Stays ON subsequently even after normal operation is restored • Synchronous with BAT. ALARM LED	S (Error)	M9007
SM52	Battery low	OFF : Normal ON : Battery low	• Same as SM51, but goes OFF subsequently when battery voltage returns to normal.	S (Error)	M9006
SM53	AC DOWN detection	OFF : AC DOWN detected ON : AC DOWN not detected	• Comes ON when there is a momentary power interruption not exceeding 20 ms; reset by turning the power OFF then ON again.	S (Error)	M9005
SM54	MINI link errors	OFF : Normal ON : Error	• Goes ON if MINI (S3) link error is detected at even one of the installed AJ71PT32 (S3) modules. Stays ON subsequently even after normal operation is restored.	S (Error)	M9004
SM56	Operation Errors	OFF : Normal ON : Operation error	• ON when operation error is generated. • Stays ON subsequently even if normal operations restored	S (Error)	M9011
SM60	Blown fuse detection	OFF : Normal ON : Module with blown fuse	• Comes ON even if there is only one output module with a blown fuse, and remains ON even after return to normal. • Blown fuse state is checked even for remote I/O station output modules.	S (Error)	M9000
SM61	I/O module Verification error	OFF : Normal ON : Error	• Comes ON if there is a discrepancy between the actual I/O modules and the registered information when the power is turned on. • I/O module verification is also conducted for remote I/O station modules.	S (Error)	M9002
SM62	Annunciator detection	OFF : Not detected ON : Detected	• Goes ON if even one annunciator F goes ON.	S (Instruction execution)	M9009
SM80	CHK detection	OFF : Not detected ON : Detected	• Goes ON if error is detected by CHK instruction. • Stays ON subsequently even after normal operation is restored.	S (Instruction execution)	New
SM90	Startup of watchdog timer for step transition (Enabled only when SFC program exists)	OFF : Not started (watchdog timer reset) ON : Started (watchdog timer started)	Corresponds to SD90	• Goes ON when measurement of step transition watchdog timer is commenced. • Resets watchdog timer when it goes OFF.	U	M9108
SM91			Corresponds to SD91			M9109
SM92			Corresponds to SD92			M9110
SM93			Corresponds to SD93			M9111
SM94			Corresponds to SD94			M9112
SM95			Corresponds to SD95			M9113
SM96			Corresponds to SD96			M9114
SM97			Corresponds to SD97			New
SM98			Corresponds to SD98			New
SM99			Corresponds to SD99			New

Special Relay List

(2) System information

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU M9□□□
SM202	LED off command	OFF → ON : LED off	• At change from OFF to ON, the LEDs corresponding to the individual bits at SD202 go off	U	New
SM203	STOP contact	STOP state	• Goes ON at STOP state	S (Status change)	M9042
SM204	PAUSE contact	PAUSE state	• Goes ON at PAUSE state	S (Status change)	M9041
SM205	STEP-RUN contact	STEP-RUN state	• Goes ON at STEP-RUN state	S (Status change)	M9054
SM206	PAUSE enable coil	OFF : PAUSE disabled ON : PAUSE enabled	• PAUSE state is entered if this relay is ON when the remote PAUSE contact goes ON	U	M9040
SM210	Clock data set request	OFF : Ignored ON : Set request	• When this relay goes from OFF to ON, clock data being stored from SD210 through SD213 after execution of END instruction for changed scan is written to the clock device.	U	M9025
SM211	Clock data error	OFF : No error ON : Error	• ON when error is generated in clock data (SD210 through SD213) value, and OFF if no error is detected.	S (Request)	M9026
SM212	Clock data display	OFF : Ignored ON : Display	• Displays clock data as month, day, hour, minute, and second at the LED display at front of CPU. (Enabled only for Q3ACPU and Q4ACPU)	U	M9027
SM213	Clock data read request	OFF : Ignored ON : Read request	• When this relay is ON, clock data is read to SD210 through SD213 as BCD values.	U	M9028
SM250	Max. loaded I/O read	OFF : Ignored ON : Read	• When this relay goes from OFF to ON, maximum loaded I/O number is read to SD250.	U	New
SM251	I/O change flag	OFF : No replacement ON : Replacement	• After the head I/O number of the I/O module being replaced is set in SD251 is set, on-line I/O module replacement is enabled when this relay is ON. (Only one module can be replaced at each setting.) • To replace an I/O module in the RUN state, use the program or a peripheral device to turn this relay ON; to replace an I/O module in the STOP state, turn this relay ON in the test mode of a peripheral device. • Do not switch between RUN and STOP states until I/O module replacement is completed.	U (END)	M9094
SM252	I/O change OK	OFF : Replacement prohibited ON : Replacement enabled	• Goes ON when I/O replacement is OK.		
SM255	MELSECNET/10 module 1 information	OFF : Operative network ON : Standby network	• Goes ON for standby network (If no designation has been made concerning active or standby, active is assumed.)	S (Initial)	New
SM256		OFF : Reads ON : Does not read	• For refresh from link to CPU (B, W, etc.) indicate whether to read from the link module.	U	New
SM257		OFF : Writes ON : Does not write	• For refresh from CPU to link (B, W, etc.), designate whether to write to the link module.	U	New
SM260	MELSECNET/10 module 2 information	OFF : Operative network ON : Standby network	• Goes ON for standby network (If no designation has been made concerning active or standby, active is assumed.)	S (Initial)	New
SM261		OFF : Reads ON : Does not read	• For refresh from link to CPU (B, W, etc.) indicate whether to read from the link module.	U	New
SM262		OFF : Writes ON : Does not write	• For refresh from CPU to link (B, W, etc.), designate whether to write to the link module.	U	New
SM265	MELSECNET/10 module 3 information	OFF : Operative network ON : Standby network	• Goes ON for standby network (If no designation has been made concerning active or standby, active is assumed.)	S (Initial)	New
SM266		OFF : Reads ON : Does not read	• For refresh from link to CPU (B, W, etc.) indicate whether to read from the link module.	U	New
SM267		OFF : Writes ON : Does not write	• For refresh from CPU to link (B, W, etc.), designate whether to write to the link module.	U	New
SM270	MELSECNET/10 module 4 information	OFF : Operative network ON : Standby network	• Goes ON for standby network (If no designation has been made concerning active or standby, active is assumed.)	S (Initial)	New
SM271		OFF : Reads ON : Does not read	• For refresh from link to CPU (B, W, etc.) indicate whether to read from the link module.	U	New
SM272		OFF : Writes ON : Does not write	• For refresh from CPU to link (B, W, etc.), designate whether to write to the link module.	U	New

Special Relay List (Continued)

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU M9□□□
SM320	Presence /absence of SFC program	OFF : SFC program absent ON : SFC program present	• ON if SFC program is correctly registered, and OFF if not registered. • Goes OFF if SFC dedicated instruction is not correct.	S (Initial)	M9100
SM321	Start/stop SFC program	OFF : SFC program stop ON : SFC program start	• Initial value is set at the same value as SM900. (Goes ON automatically if SFC program is present.) • SFC program will not execute if this goes OFF prior to SFC program processing. • Subsequently, starts SFC program when this goes from OFF to ON. • Subsequently, stops SFC program when this goes from ON to OFF.	S (Initial) U	M9101 format change
SM322	SFC program start state	OFF : Initial start ON : Restart	• Initial value is set at ON or OFF depending on parameters. • When OFF, all execution states are cleared from time SFC program was stopped; starts from the initial step of block where the start request was made. • When ON, starts from execution block and execution step active at time SFC program was stopped. (ON is enabled only when resumptive start has been designated at parameters.) • SM902 is not automatically designated for latch.	S (Initial) U	M9102 format change
SM323	Presence /absence of continuous transition for entire block	OFF : Continuous transition not effective ON : Continuous transition effective	• When OFF, transition occurs at one scan/one step, for all blocks. • When ON, transition occurs continuously for all blocks in one scan. • In designation of individual blocks, priority is given to the continuous transition bit of the block. (Designation is checked when block starts.)	U	M9103
SM324	Continuous transition prevention flag	OFF : When transition is executed ON : When no transition	• When continuous transition is effective, goes ON when continuous transition is not being executed; goes OFF when continuous transition is being executed. • Normally ON when continuous transition is not effective.	S (Instruction execution)	M9104
SM325	Output mode at block stop	OFF : OFF ON : Preserves	When block stops, selects active step operation output. • All coil outputs go OFF when OFF. • Coil outputs are preserved when ON.	S (Status change)	M9196

Special Relay List

(3) System clocks/counters

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU M9□□□
SM400	Always ON	ON OFF	• Normally is ON	S (Every END processing)	M9036
SM401	Always OFF	ON OFF	• Normally is OFF	S (Every END processing)	M9037
SM402	ON for 1 scan only after RUN	ON OFF 1 scan	• After RUN, ON for one scan only. • This connection can be used for scan execution type programs only.	S (Every END processing)	M9038
SM403	After RUN, OFF for 1 scan only	ON OFF 1 scan	• After RUN, OFF for 1 scan only • This connection can be used for scan execution type programs only.	S (Every END processing)	M9039
SM404	ON for 1 scan only after RUN	ON OFF 1 scan	• After RUN, ON for one scan only. • This contact can be used for low-speed execution type programs only.	S (Every END processing)	New
SM405	After RUN, OFF for 1 scan only	ON OFF 1 scan	• After RUN, OFF for 1 scan only • This contact can be used for low-speed execution type programs only.	S (Every END processing)	New
SM410	0.1 second clock	0.05 sec. 0.05 sec.	• Repeatedly changes between ON and OFF at each designated time interval. • Operation continues even during STOP. • When power supply is turned OFF, or reset is performed, goes from OFF to start.	S (Status change)	M9030
SM411	0.2 second clock	0.1 sec. 0.1 sec.			M9031
SM412	1 second clock	0.5 sec. 0.5 sec.			M9032
SM413	2 second clock	1 sec. 1 sec.			M9033
SM414	2n second clock	n sec. n sec.	• Goes between ON and OFF in accordance with the number of seconds designated by SD414.		M9034 format change
SM420	User timing clock No.0	n2 scan n2 scan n1 scan	• Relay repeats ON/OFF switching at fixed scan intervals. • When power supply is turned ON, or reset is performed, goes from OFF to start. • The ON/OFF intervals are set with the DUTY instruction. DUTY n1 n2 SM420	S (Every END processing)	M9020
SM421	User timing clock No.1				M9021
SM422	User timing clock No.2				M9022
SM423	User timing clock No.3				M9023
SM424	User timing clock No.4				M9024
SM430	User timing clock No.5		• For use with SM420 through SM424 low speed programs.	S (Every END processing)	New
SM431	User timing clock No.6				
SM432	User timing clock No.7				
SM433	User timing clock No.8				
SM434	User timing clock No.9				

Special Relay List

(4) Scan information

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU M9□□□
SM510	Low speed program execution flag	OFF : Completed or not executed ON : Execution under way.	• Goes ON when low-speed execution type program is executed.	S (Every END processing)	New
SM551	Reads module service interval	OFF : Ignored ON : Read	• When this goes from OFF to ON, the module service interval designated by SD550 is read to SD551 through 552.	U	New

(5) Memory cards

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU M9□□□
SM600	Memory card A usable flags	OFF : Unusable ON : Use enabled	• ON when memory card A is ready for use by user	S (Initial)	New
SM601	Memory card A protect flag	OFF : No protect ON : Protect	• Goes ON when memory card A protect switch is ON	S (Initial)	New
SM602	Drive 1 flag	OFF : No drive 1 ON : Drive 1 present	• Goes ON when drive 1 (card 1 RAM area) is present	S (Initial)	New
SM603	Drive 2 flag	OFF : No drive 2 ON : Drive 2 present	• Goes ON when drive 2 (card 1 ROM area) is present	S (Initial)	New
SM604	Memory card A in-use flag	OFF : Not in use ON : In use	• Goes ON when memory card A is in use	S (Initial)	New
SM605	Memory card A remove/insert prohibit flag	OFF : Remove/insert enabled ON : Remove/insert prohibited	• Goes ON when memory card A cannot be inserted or removed	U	New
SM620	Memory card B usable flags	OFF : Unusable ON : Use enabled	• ON when memory card B is ready for use by user	S (Initial)	New
SM621	Memory card B protect flag	OFF : No protect ON : Protect	• Goes ON when memory card B protect switch is ON	S (Initial)	New
SM622	Drive 3 flag	OFF : No drive 3 ON : Drive 3 present	• Goes ON when drive 3 (card 2 RAM area) is present	S (Initial)	New
SM623	Drive 4 flag	OFF : No drive 4 ON : Drive 4 present	• Goes ON when drive 4 (card 2 ROM area) is present	S (Initial)	New
SM624	Memory card B in-use flag	OFF : Not in use ON : n use	• Goes ON when memory card B is in use	S (Initial)	New
SM625	Memory card B remove/insert prohibit flag	OFF : Remove/insert enabled ON : Remove/insert prohibited	• Goes ON when memory card B cannot be inserted or removed	U	New
SM640	File register use	OFF : File register not in use ON : File register in use	• Goes ON when file register is in use	S (Status change)	New
SM650	Comment use	OFF : Comment not used ON : Comment in use	• Goes ON when comment file is in use	S (Status change)	New
SM660	Boot operation	OFF : Internal memory execution ON : Boot operation in progress	• Goes ON while boot operation is in process • Goes OFF if boot designation switch is OFF	S (Status change)	New
SM672	Memory card A file register access range flag	OFF : Within access range ON : Outside aocess range	• Goes ON when access is made to area outside the range of file register R of memory card A (Set within END processing.) • Reset at user program	S/U	New
SM673	Memory card B file register access range flag	OFF : Within access range ON : Outside access range	• Goes ON when access is made outside the range of file registers, R. of memory card B. (Set within END processing.) • Reset at user program	S/U	New

Special Relay List

(6) Instruction-Related Special Relays

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU M9□□□
SM700	Carry flag	OFF : Carry OFF ON : Carry ON	• Carry flag used in application instruction	S (Instruction execution)	M9012
SM701	Number of output characters selection	OFF : 16 characters output ON : Outputs until NUL	• When SM701 is OFF, 16 characters of ASCII code are output. • When SM701 is ON, output conducted until NUL (00H) code is encountered.	U	M9049
SM702	Search method	OFF : Search next ON : 2-part search	• Designates method to be used by search instruction. • Data must be arranged for 2-part search.	U	New
SM703	Sort order	OFF : Ascending order ON : Descending order	• The sort instruction is used to designate whether data should be sorted in ascending order or in descending order.	U	New
SM704	Block comparison	OFF : Non-match found ON : All match	• Goes ON when all data conditions have been met for the BKCMP instruction.	S (Instruction execution)	New
SM710	CHK instruction priority ranking flag	OFF : Conditions priority ON : Pattern priority	• Remains as originally set when OFF. • CHK priorities updated when ON.	S (Instruction execution)	New
SM711	Divided transmission status	OFF : Other than during divided processing ON : During divided processing	• In processing of AD57(S1), goes ON when screen is split for transfer, and goes OFF when split processing is completed.	S (Instruction execution)	M9065
SM712	Transmission processing selection	OFF : Batch transmission ON : Divided transmission	• In processing of AD57(S1), goes ON when canvas screen is divided for transfer.	S (Instruction execution)	M9066
SM714	Communication request registration area BUSY signal	OFF : Communication request to remote terminal module enabled ON : Communication request to remote terminal module disabled	• Used to determine whether communications requests to remote terminal modules connected to the AJ71PT32-S3 or A2CCPU can be executed or not.	S (Instruction execution)	M9081
SM715	EI flag	0 : During DI 1 : During EI	ON when EI instruction is being executed.	S (Instruction execution)	New
SM736	PKEY instruction execution in progress flag	OFF : Instruction not executed ON : Instruction execution	• ON when PKEY instruction is being executed. Goes OFF when CR is input, or when input character string reaches 32 characters.	S (Instruction execution)	New
SM737	Keyboard input reception flag for PKEY instruction	OFF : Keyboard input reception enabled ON : Keyboard input reception disabled	• Goes ON when keyboard input is being conducted. Goes when keyboard input has been stored at the CPU.	S (Instruction execution)	New
SM738	MSG instruction reception flag	OFF : Instruction not executed ON : Instruction execution	• Goes ON when MSG instruction is executed.	S (Instruction execution)	New
SM774	PID bumpless processing	OFF : Forces match ON : Does not force match	• In manual mode, designates whether or not to force the SV value to match the PV value.	U	New
SM775	Selection of link refresh processing during COM instruction execution	OFF : Performs link refresh ON : No link refresh performed	• Select whether or not to perform link refresh processing in cases where only general data processing will be conducted during the execution of the COM instruction.	U	New

(7) Debug

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU M9□□□
SM800	Sampling trace preparation	OFF : Not prepared ON : Ready	• Goes ON when sampling trace is ready	S (Status change)	New
SM801	Sampling trace start	OFF : Suspend ON : Start	• Sampling trace started when this goes ON • Suspended when OFF (Related special M all OFF)	U	M9047
SM802	Sampling trace execution in progress	OFF : Suspend ON : Start	• Goes ON during execution of sampling trace	S (Status change)	M9046
SM803	Sampling trace trigger	OFF → ON : Start	• Sampling trace trigger goes ON when this goes from OFF to ON (Identical to STRA instruction execution state)	U	M9044
SM804	After sampling trace trigger	OFF : Not after trigger ON : After trigger	• Goes ON after sampling trace trigger	S (Status change)	New
SM805	Sampling trace completed	OFF : Not completed ON : End	• Goes ON at completion of sampling trace	S (Status change)	9043
SM806	Status latch preparation	OFF : Not prepared ON : Ready	• Goes ON when status latch is ready	S (Status change)	New
SM807	Status latch command	OFF → ON : Latch	• Runs status latch command	U	New
SM808	Status latch completion	OFF : Latch not completed ON : Latch completed	• Comes ON when status latch is completed.	S (Status change)	9055
SM809	Status latch clear	OFF → ON : Clear	• Enable next status latch	U	New
SM810	Program trace preparation	OFF : Not ready ON : Ready	• Goes ON when program trace is ready	S (Status change)	New
SM811	Start program trace	OFF : Suspend ON : Start	• Program trace started when this goes ON • Suspended when OFF (Related special relays all OFF)	U	New
SM812	Program trace execution underway	OFF : Suspend ON : Start	• ON when program trace execution is underway	S (Status change)	New
SM813	Program trace trigger	OFF → ON : Start	• Program trace trigger goes ON when this goes from OFF to ON (Identical to PTRA instruction execution status)	U	New
SM814	After program trace trigger	OFF : Not after trigger ON : After trigger	• Goes ON after program trace trigger	S (Status change)	New
SM815	Program trace completion	OFF : Not completed ON : End	• Goes ON at completion of program trace	S (Status change)	New
SM820	Step trace preparation	OFF : Not prepared ON : Ready	• Goes ON after program trace registration, at ready.	S (Status change)	New
SM821	Step trace starts	OFF : Suspend ON : Start	• When this goes ON, step trace is started • Suspended when OFF (Related special relays all OFF)	U	M9182 format change
SM822	Step trace execution underway	OFF : Suspend ON : Start	• Goes ON when step trace execution is underway • Goes OFF at completion or suspension	S (Status change)	M9181
SM823	After step trace trigger	OFF : Not after trigger ON : Is after first trigger	• Goes ON if even 1 block within the step trace being executed is triggered. • Goes OFF when step trace is commenced.	S (Status change)	New
SM824	Step trace After trigger	OFF : Is not after all triggers ON : Is after all triggers	• Goes ON if all blocks within the step trace being executed are triggered. • Goes OFF when step trace is commenced.	S (Status change)	New
SM825	Step trace completed	OFF : Not completed ON : End	• Goes ON at step trace completion. • Goes OFF when step trace is commenced.	S (Status change)	M9180
SM826	Sampling trace error	OFF : Normal ON : Errors	• Goes ON if error occurs during execution of sampling trace.	S (Status change)	New
SM827	Status latch error	OFF : Normal ON : Errors	• Goes ON if error occurs during execution of status latch.	S (Status change)	New
SM828	Program trace error	OFF : Normal ON : Errors	• Goes ON if error occurs during execution of program trace.	S (Status change)	New

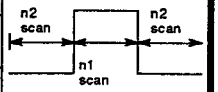
(8) Latch area

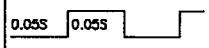
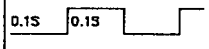
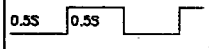
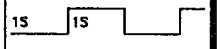
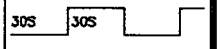
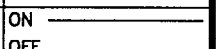
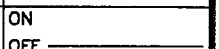
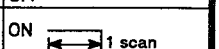
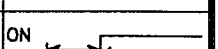
Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU M9□□□
SM900	Power cut file	OFF : No power cut file ON : Power cut file present	• Goes ON if a file is present during access when power is interrupted.	S/U (Status change)	New
SM910	RKEY registration flag	OFF : Keyboard input not registered ON : Keyboard input registered	• Goes ON at registration of keyboard input. • OFF if keyboard input is not registered.	S (Instruction execution)	New

(9) A to QnA conversion correspondences

The special relays obtained when ACPU special relays M1000 through M1255 are subjected to A - QnA conversion are described here.
These special relays are all set by the system, and cannot be turned ON or OFF with user programs.
Users who wish to turn these relays ON or OFF should edit the programs at the special relays for QnA.
However, of the special relays from SM1200 to SM1255, only those whose M9200 to M9255 equivalent before conversion could be turned ON and OFF by the user will be able to be turned ON and OFF by the user after conversion.
For details on the ACPU special relays, see the user's manuals for the individual CPUs and the MELSECNET and MELSECNET/B data link system reference manuals.

Special Relay List

ACPU Special Relay	Special Relay after Conversion	Special Relay for Modification	Name	Meaning
M9000	SM1000	SM60	Fuse blown	OFF : Normal ON : Fuse blown module with blown fuse present
M9002	SM1002	SM61	I/O module verification error	OFF : Normal ON : Error
M9004	SM1004	SM54	MINI link error	OFF : Normal ON : Error
M9005	SM1005	SM53	AC DOWN detection	OFF : AC DOWN not detected ON : AC DOWN detected
M9006	SM1006	SM52	Battery low	OFF : Normal ON : Battery low
M9007	SM1007	SM51	Battery low latch	OFF : Normal ON : Battery low
M9008	SM1008	SM1	Self-diagnostic error	OFF : No error ON : Error
M9009	SM1009	SM62	Annunciator detection	OFF : No F number detected ON : F number detected
M9010	SM1010	SM56	Operation error flag	OFF : No error ON : Error
M9012	SM1012	SM700	Carry flag *	—
M9016	SM1016	—	Data memory clear flag	OFF : Ignored ON : Output cleared
M9017	SM1017	—	Data memory clear flag	OFF : Ignored ON : Output cleared
M9020	SM1020	SM420	User timing clock No.0	
M9021	SM1021	SM421	User timing clock No.1	
M9022	SM1022	SM422	User timing clock No.2	
M9023	SM1023	SM423	User timing clock No.3	
M9024	SM1024	SM424	User timing clock No.4	
M9025	SM1025	SM210	Clock data set request	OFF : Ignored ON : Set request present used
M9026	SM1026	SM211	Clock data error	OFF : No error ON : Error

ACPU Special Relay	Special Relay after Conversion	Special Relay for Modification	Name	Meaning
M9027	SM1027	SM212	Clock data display	OFF : Ignored ON : Display
M9028	SM1028	SM213	Clock data read request	OFF : Ignored ON : Read request
M9029	SM1029	—	Batch processing of data communications requests	OFF : Batch processing not conducted ON : Batch processing conducted
M9030	SM1030	SM410	0.1 second clock	
M9031	SM1031	SM411	0.2 second clock	
M9032	SM1032	SM412	1 second clock	
M9033	SM1033	SM413	2 second clock	
M9034	SM1034	SM414	1 minute clock	
M9036	SM1036	SM400	Always ON	
M9037	SM1037	SM401	Always OFF	
M9038	SM1038	SM402	ON for 1 scan only after RUN	
M9039	SM1039	SM403	RUN flag (After RUN, OFF for 1 scan only)	
M9040	SM1040	SM206	PAUSE enable coil	OFF : PAUSE disabled ON : PAUSE enabled
M9041	SM1041	SM204	PAUSE status contact	OFF : PAUSE not in effect ON : PAUSE in effect
M9042	SM1042	SM203	STOP status contact	OFF : STOP not in effect ON : STOP in effect

* Since the SM1012 after conversion is not controlled by the system, it must be modified by using SM700.

Special Relay List (Continued)

ACPU Special Relay	Special Relay after Conversion	Special Relay for Modification	Name	Meaning
M9043	SM1043	SM805	Sampling Trace completed	OFF : Sampling trace in progress ON : Sampling trace completed
M9044	SM1044	SM803	Sampling trace	0 → 1 STRA Same as execution 1 → 0 STRAR Same as execution
M9045	SM1045	—	Watchdog timer (WDT) reset	OFF : Does not reset WDT ON : Resets WDT
M9046	SM1046	SM802	Sampling trace	OFF : Trace not in progress ON : Trace in progress
M9047	SM1047	SM801	Sampling trace preparations	OFF : Sampling trace suspended ON : Sampling trace started
M9049	SM1049	SM701	Selection of number of characters output	OFF : Output until NUL code encountered ON : 16 characters output
M9051	SM1051	—	CHG instruction execution disable	OFF : Enabled ON : Disable
M9052	SM1052	—	SEG instruction switch	OFF : 7SEC segment display ON : I/O partial refresh
M9054	SM1054	SM205	STEP RUN flag	OFF : STEP RUN not in effect ON : STEP RUN in effect
M9055	SM1055	SM808	Status latch completion flag	OFF : Not completed ON : Completed
M9056	SM1056	—	Main side P, I set request	OFF : Other than when P, I set being requested ON : P, I set being requested
M9057	SM1057	—	Sub side P, I set request	OFF : Other than when P, I set being requested ON : P, I set being requested
M9058	SM1058	—	Main program P, I set completion	Momentarily ON at P, I set completion
M9059	SM1059	—	Sub program P, I set completion	Momentarily ON at P, I set completion
M9060	SM1060	—	Sub program 2 P, I set request	OFF : Other than when P, I set being requested ON : P, I set being requested
M9061	SM1061	—	Sub program 3 P, I set request	OFF : Other than when P, I set being requested ON : P, I set being requested
M9065	SM1065	SM711	Divided processing execution detection	OFF : Divided processing not underway ON : During divided processing
M9066	SM1066	SM712	Divided processing request flag	OFF : Batch processing ON : Divided processing

ACPU Special Relay	Special Relay after Conversion	Special Relay for Modification	Name	Meaning
M9070	SM1070	—	A8UPU/A8PUJ required search time	OFF : Read time not shortened ON : Read time shortened
M9081	SM1081	SM714	Communication request registration area BUSY signal	OFF : Empty spaces in communication request registration area ON : No empty spaces in communication request registration area
M9084	SM1084	—	Error check	OFF : Error check executed ON : No error check
M9091	SM1091	—	Instruction error flag	OFF : No error ON : Error
M9094	SM1094	SM251	I/O change flag	OFF : Replacement ON : No replacement
M9100	SM1100	SM320	Presence/ absence of SFC program	OFF : SFC programs not used ON : SFC programs used
M9101	SM1101	SM321	Start/stop SFC program	OFF : SFC programs stop ON : SFC programs start
M9102	SM1102	SM322	SFC program start state	OFF : Initial Start ON : Continue
M9103	SM1103	SM323	Presence/ absence of continuous transition	OFF : Continuous transition not effective ON : Continuous transition effective
M9104	SM1104	SM324	Continuous transition suspension flag	OFF : When transition is completed ON : When no transition
M9108	SM1108	SM90	Step transition watchdog timer start (equivalent of D9108)	OFF : Watchdog timer reset ON : Watchdog timer reset start
M9109	SM1109	SM91	Step transition watchdog timer start (equivalent of D9109)	
M9110	SM1110	SM92	Step transition watchdog timer start (equivalent of D9110)	
M9111	SM1111	SM93	Step transition watchdog timer start (equivalent of D9111)	
M9112	SM1112	SM94	Step transition watchdog timer start (equivalent of D9112)	
M9113	SM1113	SM95	Step transition watchdog timer start (equivalent of D9113)	
M9114	SM1114	SM96	Step transition watchdog timer start (equivalent of D9114)	

Special Relay List (Continued)

ACPU Special Relay	Special Relay after Conversion	Special Relay for Modification	Name	Meaning
M9180	SM1180	SM825	Active step sampling trace completion flag	OFF : Trace started ON : Trace completed
M9181	SM1181	SM822	Active step sampling trace execution flag	OFF : Trace not being executed ON : Trace execution under way
M9182	SM1182	SM821	Active step sampling trace permission	OFF : Trace disable/suspend ON : Trace enable
M9196	SM1196	SM325	Operation output at block stop	OFF : Coil output OFF ON : Coil output ON
M9197 • M9198	SM1197 • SM1198	—	Switch between blown fuse and I/O verification error display	Display is changed depending on combination of M9197 ON/OFF state and M9198 ON/OFF state.
M9199	SM1199	—	On-line recovery of sampling trace status latch data	OFF : Does not perform data recovery ON : Performs data recovery
M9200	SM1200	—	LRDP instruction reception	OFF : Not accepted ON : Accepted
M9201	SM1201	—	LRDP instruction completion	OFF : Not completed ON : End
M9202	SM1202	—	LWTP instruction reception	OFF : Not accepted ON : Accepted
M9203	SM1203	—	LWTP instruction completion	OFF : Not completed ON : End
M9204	SM1204	—	LRDP instruction completion	OFF : Not completed ON : End
M9205	SM1205	—	LWTP instruction completion	OFF : Not completed ON : End
M9206	SM1206	—	Host station link parameter error	OFF : Normal ON : Abnormal
M9207	SM1207	—	Link parameter check results	OFF : YES ON : NO
M9208	SM1208	—	Sets master station B and W transmission range (for lower link master stations only).	OFF : Transmits to tier 2 and tier 3 ON : Transmits to tier 2 only
M9209	SM1209	—	Link parameter check command (for lower link master stations only).	OFF : Executing the check function ON : Check non-execution
M9210	SM1210	—	Link card error (for local station)	OFF : Normal ON : Abnormal
M9211	SM1211	—	Link module error (for master station use)	OFF : Normal ON : Abnormal
M9224	SM1224	—	Link state	OFF : Online ON : Offline, station-to-station test, or self-loopback test
M9225	SM1225	—	Forward loop error	OFF : Normal ON : Abnormal
M9226	SM1226	—	Reverse loop error	OFF : Normal ON : Abnormal

ACPU Special Relay	Special Relay after Conversion	Special Relay for Modification	Name	Meaning
M9227	SM1227	—	Loop test state	OFF : Not being executed ON : Forward or reverse loop test execution underway
M9232	SM1232	—	Local station operation state	OFF : RUN or STEP RUN state ON : STOP or PAUSE state
M9233	SM1233	—	Local station error detect state	OFF : No errors ON : Error detection
M9235	SM1235	—	Local station, remote I/O station parameter error detect state	OFF : No errors ON : Error detection
M9236	SM1236	—	Local station, remote I/O station initial communications state	OFF : No communications ON : Communications underway
M9237	SM1237	—	Local station, remote I/O station error	OFF : Normal ON : Abnormal
M9238	SM1238	—	Local station, remote I/O station forward or reverse loop error	OFF : Normal ON : Abnormal
M9240	SM1240	—	Link state	OFF : Online ON : Offline, station-to-station test, or self-loopback test
M9241	SM1241	—	Forward loop line error	OFF : Normal ON : Abnormal
M9242	SM1242	—	Reverse loop line error	OFF : Normal ON : Abnormal
M9243	SM1243	—	Loopback implementation	OFF : Loopback not being conducted ON : Loopback implementation
M9246	SM1246	—	Data not received	OFF : Reception ON : No reception
M9247	SM1247	—	Data not received	OFF : Reception ON : No reception
M9250	SM1250	—	Parameters not received	OFF : Reception ON : No reception
M9251	SM1251	—	Link relay	OFF : Normal ON : Abort
M9252	SM1252	—	Loop test state	OFF : Not being executed ON : Forward or reverse loop test execution underway
M9253	SM1253	—	Master station operation state	OFF : RUN or STEP RUN state ON : STOP or PAUSE state
M9254	SM1254	—	Local station other than host station operation state	OFF : RUN or STEP RUN state ON : STOP or PAUSE state
M9255	SM1255	—	Local station other than host station error	OFF : Normal ON : Abnormal

APPENDIX 3 SPECIAL REGISTER LIST

The special registers, SD, are internal registers with fixed applications in the programmable controller.
For this reason, it is not possible to use these registers in sequence programs in the same way that normal registers are used.
However, it is possible to write data to them in order to conduct QnACPU controls.
Data stored in the special registers are stored as BIN values if no special designation has been made to the contrary.

The headings in the table that follows have the following meanings.

Item	Function of Item
Number	• Indicates special register number
Name	• Indicates name of special register
Meaning	• Indicates contents of special register
Explanation	• Discusses contents of special register in more detail
Explanation Set by (When set)	• Indicates whether the relay is set by the system or user, and, if it is set by the system, when setting is performed. <Set by> S : Set by system U : Set by user (sequence programs or test operations from peripheral devices) S/U: Set by both system and user <When set>→Indicated only for registers set by system Each END : Set during each END processing Initial : Set only during initial processing (when power supply is turned ON, or when going from STOP to RUN) Status change : Set only when there is a change in status Error : Set when error occurs Instruction execution : Set when instruction is executed Request : Set only when there is a user request (through SM, etc.)
Corresponding ACPU D9□□□	• Indicates corresponding special register in ACPU (D9□□□) (Change and notation when there has been a change in contents) • Items indicated as "New" have been newly added for QnACPU

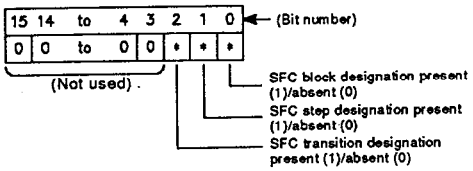
For details on the following items, see these manuals:
• Networks →MELSECNET/10 Network System Reference Manual for QnA
• SFC →QnACPU Programming Manual (SFC)

Special Register List

(1) Diagnostic Information

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9□□□
SD0	Diagnostic errors	Diagnosis error code	• Error codes for errors found by diagnosis are stored as BIN data. • The most recent error among those currently detected is stored.	S (Error)	D9008 format change
SD1	Clock time for diagnosis error occurrence	Clock time for diagnosis error occurrence	• Year (last two digits) and month that SD0 data was updated is stored as BCD 2-digit code. B15 to B8 B7 to B0 (Example) Year (0 to 99) | Month (1 to 12) : October, 1995 H9510	S (Error)	New
SD2			• The day and hour that SD0 was updated is stored as BCD 2-digit code. B15 to B8 B7 to B0 (Example) Day (1 to 31) | Hour (0 to 23) : 10 p.m. on 25th H2510		
SD3			• The minute and second that SD0 data was updated is stored as BCD 2-digit code. B15 to B8 B7 to B0 (Example) Minutes (0 to 59) | Seconds (0 to 59) : 35 min. 48 sec. (past the hour) H3548		
SD4	Error information categories	Error information category code	Category codes which help indicate what type of information is being stored in the common information areas (SD5 through SD15) and the individual information areas (SD16 through D26) are stored here. B15 to B8 B7 to B0 Individual information category codes | Common information category codes • The common information category codes store the following codes: 0 : No error 1 : Unit/module No. 2 : File name/Drive name 3 : Time (value set) 4 : Program error location • The individual information category codes store the following codes: 0 : No error 1 : (Open) 2 : File name/Drive name 3 : Time (value actually measured) 4 : Program error location 5 : Parameter number 6 : Annunciator number 7 : Check instruction malfunction number	S (Error)	New

Special Register List (Continued)

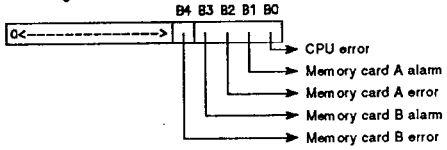
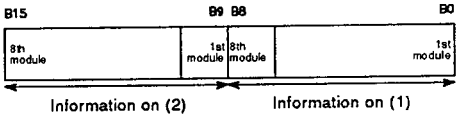
<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by
(When Set)</th><th>Corresponding
ACPU
D9□□□</th></tr>
<tr><td>SD5
SD6
SD7
SD8
SD9
SD10
SD11
SD12
SD13
SD14
SD15</td><td>Error common information</td><td>Error common information</td><td>
• Common information corresponding to the error codes (SD0) is stored here.
• The following four types of information are stored here:
(1) Unit/module No.
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD5</td><td>Unit/module No.</td></tr>
<tr><td>SD6</td><td>I/O No.</td></tr>
<tr><td>SD7
SD8
SD9
SD10
SD11
SD12
SD13
SD14
SD15</td><td>(Vacant)</td></tr>
</table>
(2) File name/Drive name
(Example)
File name=ABCDEFGH. IJK
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD5</td><td>Drive</td></tr>
<tr><td>SD6
SD7
SD8
SD9</td><td>File name
(ASCII code: 8 characters)</td></tr>
<tr><td>SD10</td><td>Extension (ASCII code: 3 characters) 2EH (.)</td></tr>
<tr><td>SD11</td><td></td></tr>
<tr><td>SD12
SD13
SD14
SD15</td><td>(Vacant)</td></tr>
</table>
<table>
<tr><th>B15 to B8</th><th>B7 to B0</th></tr>
<tr><td>B</td><td>A</td></tr>
<tr><td>D</td><td>C</td></tr>
<tr><td>F</td><td>E</td></tr>
<tr><td>H</td><td>G</td></tr>
<tr><td>I</td><td>.</td></tr>
<tr><td>K</td><td>J</td></tr>
</table>
(3) Time (value set)
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD5</td><td>Time : 1 μs units (0 to 999 μs)</td></tr>
<tr><td>SD6</td><td>Time : 1 ms units (0 to 65535 ms)</td></tr>
<tr><td>SD7
SD8
SD9
SD10
SD11
SD12
SD13
SD14
SD15</td><td>(Vacant)</td></tr>
</table>
(4) Program error location
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD5
SD6
SD7
SD8</td><td>File name
(ASCII code: 8 characters)</td></tr>
<tr><td>SD9</td><td>Extension (ASCII code: 3 characters) 2EH (.)</td></tr>
<tr><td>SD10</td><td></td></tr>
<tr><td>SD11</td><td>Pattern*</td></tr>
<tr><td>SD12</td><td>Block No.</td></tr>
<tr><td>SD13</td><td>Step No./transition No.</td></tr>
<tr><td>SD14</td><td>Sequence step No. (L)
Sequence step No. (H)</td></tr>
<tr><td>SD15</td><td></td></tr>
</table>
* Contents of pattern data


</td><td>S (Error)</td><td>New</td></tr>
</table>

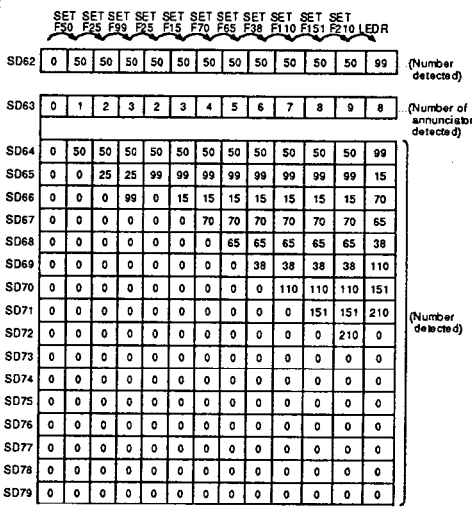
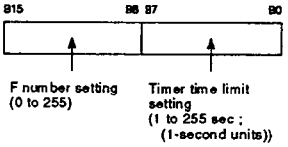
Special Register List (Continued)

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by
(When Set)</th><th>Corresponding
ACPU
D9□□□</th></tr>
<tr><td>SD16
SD17
SD18
SD19
SD20
SD21
SD22
SD23
SD24
SD25
SD26</td><td>Error individual information</td><td>Error individual information</td><td>
• Individual information corresponding to error codes (SD0) is stored here.
• The following six types of information are stored here:
(1) Unit/module No.
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD16</td><td>Unit/module No.</td></tr>
<tr><td>SD17</td><td>I/O No.</td></tr>
<tr><td>SD18
SD19
SD20
SD21
SD22
SD23
SD24
SD25
SD26</td><td>(Vacant)</td></tr>
</table>
(2) File name/Drive name
(Example)
File name=
ABCDEFGH. IJK
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD16</td><td>Drive</td></tr>
<tr><td>SD17
SD18
SD19
SD20</td><td>File name
(ASCII code: 8 characters)</td></tr>
<tr><td>SD21</td><td>Extension 2EH (.)</td></tr>
<tr><td>SD22</td><td>(ASCII code: 3 characters)</td></tr>
<tr><td>SD23
SD24
SD25
SD26</td><td>(Vacant)</td></tr>
</table>
<table>
<tr><th>B15 to B8</th><th>B7 to B0</th></tr>
<tr><td>B</td><td>A</td></tr>
<tr><td>D</td><td>C</td></tr>
<tr><td>F</td><td>E</td></tr>
<tr><td>H</td><td>G</td></tr>
<tr><td>I</td><td>.</td></tr>
<tr><td>K</td><td>J</td></tr>
</table>
(3) Time (value actually measured)
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD16</td><td>Time : 1 μs units (0 to 999 μs)</td></tr>
<tr><td>SD17</td><td>Time : 1 ms units (0 to 65535 ms)</td></tr>
<tr><td>SD18
SD19
SD20
SD21
SD22
SD23
SD24
SD25
SD26</td><td>(Vacant)</td></tr>
</table>
(4) Program error location
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD16
SD17
SD18
SD19</td><td>File name
(ASCII code: 8 characters)</td></tr>
<tr><td>SD20</td><td>Extension 2EH (.)</td></tr>
<tr><td>SD21</td><td>(ASCII code: 3 characters)</td></tr>
<tr><td>SD22</td><td>Pattern*</td></tr>
<tr><td>SD23</td><td>Block No.</td></tr>
<tr><td>SD24</td><td>Step No./transition No.</td></tr>
<tr><td>SD25
SD26</td><td>Sequence step No. (L)
Sequence step No. (H)</td></tr>
</table>
* Contents of pattern data
<table>
<tr><td>15</td><td>14</td><td>to</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>← (Bit number)</td></tr>
<tr><td>0</td><td>0</td><td>to</td><td>0</td><td>0</td><td>*</td><td>*</td><td>*</td><td></td></tr>
</table>
(Not used)
SFC block designation present (1)/absent (0)
SFC step designation present (1)/absent (0)
SFC transition designation present (1)/absent (0)
</td><td>S (Error)</td><td>New</td></tr>
</table>

Special Register List (Continued)

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9□□□
SD16 SD17 SD18 SD19 SD20 SD21 SD22 SD23 SD24 SD25 SD26	Error individual information	Error individual information	(Continued) (5) Parameter number Number / Meaning SD16 / Parameter number SD17 to SD26 / (Vacant) (6) Annunciator number / CHK instruction malfunction number Number / Meaning SD16 / No. SD17 to SD26 / (Vacant)	S (Error)	New
SD50	Error reset	Error number that performs error reset	• Stores error number that performs error reset	U	New
SD51	Battery low latch	Bit pattern indicating where battery voltage drop occurred	• All corresponding bits go ON when battery voltage drops. • Subsequently, these remain ON even after battery voltage has been returned to normal. 	S (Error)	New
SD52	Battery low	Bit pattern indicating where battery voltage drop occurred	• Same configuration as SD51 above • Subsequently, goes OFF when battery voltage is restored to normal.	S (Error)	New
SD53	AC DOWN detection	Number of times for AC DOWN	• 1 is added to the stored value each time the input voltage becomes 80 % or less of the rating while the CPU module is operating, and the value is stored in BIN code.	S (Error)	D9005
SD54	MINI link errors	Error detection state	(1) The relevant station bit goes ON when any of the Installed MINI (-S3) X(n+0)/X(n+20), X(n+6)/(n+26), X(n+7)/(n+27) or X(n+8)/X(n+28) goes ON. (2) Goes ON when communications between the installed MIMI (-S3) and the CPU are not possible. 	S (Error)	D9004 format change
SD60	Blown fuse number	Number of module with blown fuse	• Value stored here is the lowest station number of the module with the blown fuse, divided by 16.	S (Error)	D9000
SD61	I/O module verification error number	I/O module verification error module number	• The lowest number of the module where the I/O module verification number took place.	S (Error)	D9002
SD62	Annunciator number	Annunciator number	• The first annunciator number to be detected is stored here.	S (Instruction execution)	D9009
SD63	Number of annunciators	Number of annunciators	• Stores the number of annunciators searched.	S (Instruction execution)	D9124

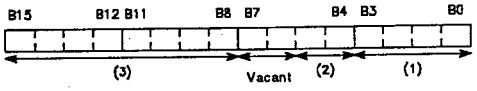
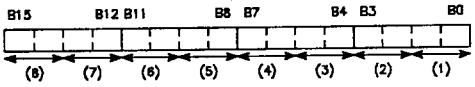
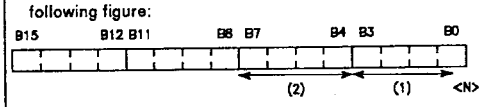
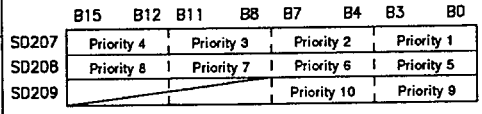
Special Register List (Continued)

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by (When Set)</th><th>Corresponding ACPU D9□□□</th></tr>
<tr><td>SD64</td><td rowspan="16">Table of detected annunciator numbers</td><td rowspan="16">Annunciator detection number</td><td rowspan="16">When F goes ON due to [OUT F] or [SET F], the F numbers which go progressively ON from SD64 through SD79 are registered.
F numbers turned OFF by [RST F] are deleted from SD64 to SD79, and are shifted to the data register following the data register where the deleted F numbers had been stored.
Execution of the [LED R] instruction shifts the contents of SD64 to SD79 up by one.
(This can also be done by using the INDICATOR RESET switch on the front of the CPU of the Q3A/Q4ACPU.)
After 16 annunciators have been detected, detection of the 17th will not be stored from SD64 through SD79.
 
</td><td rowspan="16">S (Instruction execution)</td><td>D9125</td></tr>
<tr><td>SD65</td><td>D9126</td></tr>
<tr><td>SD66</td><td>D9127</td></tr>
<tr><td>SD67</td><td>D9128</td></tr>
<tr><td>SD68</td><td>D9129</td></tr>
<tr><td>SD69</td><td>D9130</td></tr>
<tr><td>SD70</td><td>D9131</td></tr>
<tr><td>SD71</td><td>D9132</td></tr>
<tr><td>SD72</td><td>New</td></tr>
<tr><td>SD73</td><td>New</td></tr>
<tr><td>SD74</td><td>New</td></tr>
<tr><td>SD75</td><td>New</td></tr>
<tr><td>SD76</td><td>New</td></tr>
<tr><td>SD77</td><td>New</td></tr>
<tr><td>SD78</td><td>New</td></tr>
<tr><td>SD79</td><td>New</td></tr>
<tr><td>SD80</td><td>CHK number</td><td>CHK number</td><td>• Error codes detected by the CHK instruction are stored as BCD code.</td><td>S (Instruction execution)</td><td>New</td></tr>
<tr><td>SD90</td><td rowspan="10">Step transition watchdog timer setting value (Enabled only when SFC program exists)</td><td rowspan="10">F number for timer set value and time over error</td><td>Corresponds to SM90</td><td rowspan="10">• F numbers which go ON at step transition watchdog timer set value and watchdog timer over errors.
 

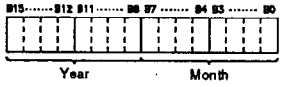
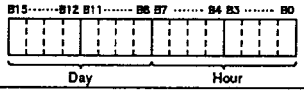
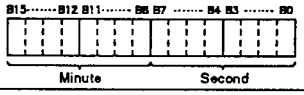
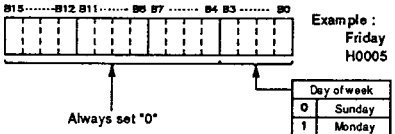
• Timer is started by turning SM90 through SM99 ON during active step, and if the transition conditions for the relevant steps are not met within the timer limits, the designated annunciator (F) will go ON.</td><td rowspan="10">U</td><td>D9108</td></tr>
<tr><td>SD91</td><td>Corresponds to SM91</td><td>D9109</td></tr>
<tr><td>SD92</td><td>Corresponds to SM92</td><td>D9110</td></tr>
<tr><td>SD93</td><td>Corresponds to SM93</td><td>D9111</td></tr>
<tr><td>SD94</td><td>Corresponds to SM94</td><td>D9112</td></tr>
<tr><td>SD95</td><td>Corresponds to SM95</td><td>D9113</td></tr>
<tr><td>SD96</td><td>Corresponds to SM96</td><td>D9114</td></tr>
<tr><td>SD97</td><td>Corresponds to SM97</td><td>New</td></tr>
<tr><td>SD98</td><td>Corresponds to SM98</td><td>New</td></tr>
<tr><td>SD99</td><td>Corresponds to SM99</td><td>New</td></tr>
</table>

Special Register List

(2) System information

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9□□□
SD200	State of switch	State of CPU switch	• The CPU switch state is stored in the following format:  (1): CPU key State of switch — 0: RUN, 1: STOP, 2: L.CLR (2): Memory cards switch — B4 corresponds to card A, and B5 corresponds to card B; OFF at 0; ON at 1 (3): DIP switch — B8 through B15 correspond to SW1 through SW8; OFF at 0; ON at 1	S (Every END processing)	New
SD201	LED status	State of CPU-LED	• Information concerning which of the following states the LEDs on the CPU are in is stored in the following bit patterns: • 0 is off, 1 is on, and 2 is flicker  (1): RUN (2): ERROR (3): USER (4): BAT.ALARM (5): BOOT (6): CARD A (Memory card) (7): CARD B (Memory card) (8): Vacant	S (Status change)	New
SD202	LED off	Bit pattern of LED that is turned off	• Stores bit patterns of LEDs turned off (Only USER and BOOT enabled) • Turned off at 1, not turned off at 0	U	New
SD203	Operating state of CPU	Operating state of CPU	• The CPU operating state is stored as indicated in the following figure:  (1): Operating state of CPU — 0: RUN, 1: STEP-RUN, 2: STOP, 3: PAUSE (2): STOP/PAUSE cause — 0: Key switch, 1: Remote contact, 2: Peripheral, computer link, or operation from some other remote source, 3: Internal program instruction, 4: Errors Note: Priority is earliest first	S (Every END processing)	D9015 format change
SD207	LED display priority ranking	Priorities 1 to 4	• When error is generated, the LED display (flicker) is made according to the error number setting priorities. • The setting areas for priorities are as follows:  Default Value SD207=H4321 SD208=H8765 SD209=H00A9 • No display is made if "0" is set. However, even if "0" has been set, information concerning CPU operation stop (including parameter settings) errors will be indicated by the LEDs without conditions.	U	D9038
SD208		Priorities 5 to 8			D3039 format change
SD209		Priorities 9 to 10			New

Special Register List (Continued)

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9□□□
SD210	Clock data	Clock data (year, month)	• The year (last two digits) and month are stored as BCD code at SD210 as shown below:  Example : July 1993 H9307	S/U (Request)	D9025
SD211	Clock data	Clock data (day, hour)	• The day and hour are stored as BCD code at SD211 as shown below:  Example : 31st, 10 a.m. H3110		D9026
SD212	Clock data	Clock data (minute, second)	• The minutes and seconds (after the hour) are stored as BCD code at SD212 as shown below:  Example : 35 min., 48 sec. (after the hour) H3548		D9027
SD213	Clock data	Clock data (day of week)	• The day of the week is stored as BCD code at SD213 as shown below:  Example : Friday H0005 Day of week: 0 Sunday, 1 Monday, 2 Tuesday, 3 Wednesday, 4 Thursday, 5 Friday, 6 Saturday		D9028
SD220 SD221 SD222 SD223 SD224 SD226 SD227	LED display data	Display indicator data	• LED display ASCII data (16 characters) stored here. B15 to B8 B7 to B0 SD220: 15th character from the right / 16th character from the right SD221: 13th character from the right / 14th character from the right SD222: 11th character from the right / 12th character from the right SD223: 9th character from the right / 10th character from the right SD224: 7th character from the right / 8th character from the right SD225: 5th character from the right / 6th character from the right SD226: 3rd character from the right / 4th character from the right SD227: 1st character from the right / 2nd character from the right	S (When changed)	New
SD250	Loaded maximum I/O	Loaded maximum I/O No.	• When SM250 goes from OFF to ON, the upper 2 digits of the final I/O number plus 1 of the modules loaded are stored as BIN values.	S (Request END)	New
SD251	Head I/O No. for replacement	Head I/O number for module replacement	• Stores the upper two digits of the first I/O number of an I/O module that is removed/replaced in the online status.	U	D9094
SD254	NET/10 information	Number of modules installed	• Indicates the number of modules installed on NET/10.	S (Initial)	New
SD255		Information from 1st module: I/O No.	• NET/10 I/O number of first module installed		
SD256		Information from 1st module: Network No.	• NET/10 network number of first module installed		
SD257		Information from 1st module: Group number	• NET/10 group number of first module installed		
SD258		Information from 1st module: Station No.	• NET/10 station number of first module installed		
SD259		Information from 1st module: Standby information	• In the case of standby stations, the module number of the standby station is stored. (1 to 4)		
SD260 to SD264		Information from 2nd module	• Configuration is identical to that for the first module.		
SD265 to SD269		Information from 3rd module	• Configuration is identical to that for the first module.		
SD270 to SD274		Information from 4th module	• Configuration is identical to that for the first module.		

Special Register List (Continued)

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9□□□
SD290	Device allocation (Same as parameter contents)	Number of points allocated for X	• Stores the number of points currently set for X devices	S (Initial)	New
SD291		Number of points allocated for Y	• Stores the number of points currently set for Y devices		
SD292		Number of points allocated for M	• Stores the number of points currently set for M devices		
SD293		Number of points allocated for L	• Stores the number of points currently set for L devices		
SD294		Number of points allocated for B	• Stores the number of points currently set for B devices		
SD295		Number of points allocated for F	• Stores the number of points currently set for F devices		
SD296		Number of points allocated for SB	• Stores the number of points currently set for SB devices		
SD297		Number of points allocated for V	• Stores the number of points currently set for V devices		
SD298		Number of points allocated for S	• Stores the number of points currently set for S devices		
SD299		Number of points allocated for T	• Stores the number of points currently set for T device		
SD300		Number of points allocated for ST	• Stores the number of points currently set for ST devices		
SD301		Number of points allocated for C	• Stores the number of points currently set for C devices		
SD302		Number of points allocated for D	• Stores the number of points currently set for D devices		
SD303		Number of points allocated for W	• Stores the number of points currently set for W devices		
SD304		Number of points allocated for SW	• Stores the number of points currently set for SW devices		

(3) System clocks/counters

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9□□□
SD412	1 second counter	Number of counts in 1-second units	• Following programmable controller CPU RUN, 1 is added each second • Count repeats from 0 to 32767 to -32768 to 0	S (Status change)	D9022
SD414	2n second clock setting	2n second clock units	• Stores value n of 2n second clock (Default is 30) • Setting can be made between 1 and 32767		
SD420	Scan counter	Number of counts in each scan	• Incremented by 1 for each scan execution after the PC CPU is set to RUN. • Count repeats from 0 to 32767 to -32768 to 0	S (Every END processing)	New
SD430	Low speed scan counter	Number of counts in each scan	• Incremented by 1 for each scan execution after the PC CPU is set to RUN. • Count repeats from 0 to 32767 to -32768 to 0 • Used only for low speed execution type programs	S (Every END processing)	New

(4) Scan information

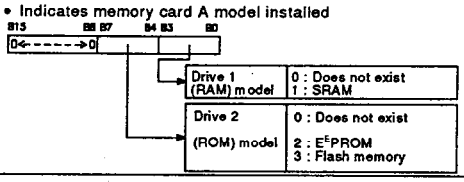
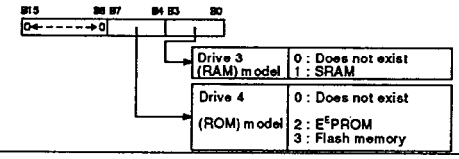
Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9□□□
SD500	Execution program No.	Execution type of program being executed	• Program number of program currently being executed is stored as BIN value.	S (Status change)	New
SD510	Low speed program No.	File name of low speed execution in progress	• Program number of low speed program currently being executed is stored as BIN value. • Enabled only when SM510 is ON.	S (Every END processing)	New
SD520	Current scan time	Current scan time (in 1 ms units)	• Stores current scan time (in 1 ms units) Range from 0 to 65535	S (Every END processing)	D9017 format change
SD521		Current scan time (in 1 μs units)	• Stores current scan time (in 1 μs units) Range from 0 to 999 (Example) A current scan of 23.6 ms would be stored as follows: D520=23 D521=600		New
SD522	Initial scan time	Initial scan time (in 1 ms units)	• Stores scan time for first scan (in 1 ms units) • Range from 0 to 65535	S (First END processing)	New
SD523		Initial scan time (in 1 μs units)	• Stores scan time for first scan (in 1 μs units) • Range of 0 to 999		
SD524	Minimum scan time	Minimum scan time (in 1 ms units)	• Stores minimum value of scan time (in 1 ms units) • Range from 0 to 65535	S (Every END processing)	D9018 format change
SD525		Minimum scan time (in 1 μs units)	• Stores minimum value of scan time (in 1 μs units) • Range of 0 to 999		New
SD526	Maximum scan time	Maximum scan time (in 1 ms units)	• Stores maximum value of scan time, excepting the first scan. (in 1 ms units) • Range from 0 to 65535	S (Every END processing)	D9019 format change
SD527		Maximum scan time (in 1 μs units)	• Stores maximum value of scan time, excepting the first scan. (in 1 μs units) • Range of 0 to 999		New
SD528	For low speed execution type programs current scan time	Current scan time (in 1 ms units)	• Stores current scan time for low speed execution type program (in 1 ms units) • Range from 0 to 65535	S (Every END processing)	New
SD529		Current scan time (in 1 μs units)	• Stores current scan time for low speed execution type program (in 1 μs units) • Range of 0 to 999		
SD532	Minimum scan time for low speed execution type programs	Minimum scan time (in 1 ms units)	• Stores minimum value of scan time for low speed execution type program (in 1 ms units) • Range from 0 to 65535	S (Every END processing)	New
SD533		Minimum scan time (in 1 μs units)	• Stores minimum value of scan time for low speed execution type program (in 1 μs units) • Range of 0 to 999		
SD534	Maximum scan time for low speed execution type programs	Maximum scan time (in 1 ms units)	• Stores the maximum scan time for all except low speed execution type program's first scan. (in 1 ms units) • Range from 0 to 65535	S (Every END processing)	New
SD535		Maximum scan time (in 1 μs units)	• Stores the maximum scan time for all except low speed execution type program's first scan. (in 1 μs units) • Range of 0 to 999		
SD540	END processing time	END processing time (in 1 ms units)	• Stores time from completion of scan program to start of next scan. (in 1 ms units) • Range from 0 to 65535	S (Every END processing)	New
SD541		END processing time (in 1 μs units)	• Stores time from completion of scan program to start of next scan. (in 100 μs units) • Range of 0 to 999		
SD542	Constant scan wait time	Constant scan wait time (in 1 ms units)	• Stores wait time when constant scan time has been set. (in 1 ms units) • Range from 0 to 65535	S (First END processing)	New
SD543		Constant scan wait time (in 1 μs units)	• Stores wait time when constant scan time has been set. (in 1 μs units) • Range of 0 to 999		
SD544	Cumulative execution time for low speed execution type programs	Cumulative execution time for low speed execution type programs (in 1 ms units)	• Stores cumulative execution time for low speed execution type programs. (in 1 ms units) • Range from 0 to 65535 • Cleared to 0 following 1 low speed scan	S (Every END processing)	New
SD545		Cumulative execution time for low speed execution type programs (in 1 μs units)	• Stores cumulative execution time for low speed execution type programs. (in 1 μs units) • Range of 0 to 999 • Cleared to 0 following 1 low speed scan		

Special Register List (Continued)

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9□□□
SD546	Execution time for low speed execution type programs	Execution time for low speed execution type programs (in 1 ms units)	• Stores low speed program execution time during 1 scan (in 1 ms units) • Range from 0 to 65535 • Stores each scan	S (Every END processing)	New
SD547		Execution time for low speed execution type programs (in 1 μs units)	• Stores low speed program execution time during 1 scan (in 1 μs units) • Range of 0 to 999 • Stores each scan		
SD548	Scan program execution time	Scan program execution time (in 1 ms units)	• Stores execution time for scan execution type program during 1 scan (in 1 ms units) • Range from 0 to 65535 • Stores each scan	S (Every END processing)	New
SD549		Scan program execution time (in 1 μs units)	• Stores execution time for scan execution type program during 1 scan (in 1 μs units) • Range of 0 to 999 • Stores each scan		
SD550	Module (unit) access interval measurement time	Unit/module No.	• Set the upper two digits of the head I/O number of the module for which the module access interval is to be measured.	U	New
SD551	Module access interval time	Module access interval (in 1 ms units)	• When SM551 is ON, stores service interval for module designated by SD550. (in 1 ms units) • Range from 0 to 65535	S (Request)	New
SD552		Module access interval (in 1 μs units)	• When SM551 is ON, stores access interval for module designated by SD550. (in 1 μs units) • Range from 0 to 999		

Special Register List

(5) Memory card

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9□□□
SD600	Memory card A models	Memory card A models	• Indicates memory card A model installed 	S (Initial)	New
SD602	Drive 1 (RAM) capacity	Drive 1 capacity	• Drive 1 capacity is stored in 1 K byte units	S (Initial)	New
SD603	Drive 2 (ROM) capacity	Drive 2 capacity	• Drive 2 capacity is stored in 1 K byte units	S (Initial)	New
SD604	Memory card A use conditions	Memory card A use conditions	• The use conditions for memory card A are stored as bit patterns (In use when ON) • The significance of these bit patterns is indicated below: B0 : Boot operation (QBT) B1 : Parameters (QPA) B2 : Device comments (QCD) B3 : Device initial value (QDI) B4 : File R (QDR) B5 : Sampling trace (QTS) B6 : Status latch (QTL) B7 : Program trace (QTP) B8 : Simulation data (QDS) B9 : CPU fault history (QFD) B10 : SFC trace (QTS) B11 : Local device (QDL) B12 : B13 : B14 : B15 :	S (Status change)	New
SD620	Memory card B models	Memory card B models	• Indicates memory card B models installed 	S (Initial)	New
SD622	Drive 3 (RAM) capacity	Drive 3 capacity	• Drive 3 capacity is stored in 1 K byte units	S (Initial)	New
SD623	Drive 4 (ROM) capacity	Drive 4 capacity	• Drive 4 capacity is stored in 1 K byte units	S (Initial)	New
SD624	Memory card B use conditions	Memory card B use conditions	• The use conditions for memory card B are stored as bit patterns (In use when ON) • The significance of these bit patterns is indicated below: B0 : Boot operation (QBT) B1 : Parameters (QPA) B2 : Device comments (QCD) B3 : Device initial value (QDI) B4 : File R (QDR) B5 : Sampling trace (QTS) B6 : Status latch (QTL) B7 : Program trace (QTP) B8 : Simulation data (QDS) B9 : CPU fault history (QFD) B10 : SFC trace (QTS) B11 : Local device (QDL) B12 : B13 : B14 : B15 :		
SD640	File register drive	Drive number:	• Stores drive number being used by file register	S (Status change)	New
SD641 SD642 SD643 SD644 SD645 SD646	File register file name	File register file name	• Stores file register file name (with extension) selected at parameters or by use of QDRSET instruction as ASCII code. SD641: B15 to B8 = Second character; B7 to B0 = First character SD642: Fourth character; Third character SD643: Sixth character; Fifth character SD644: Eighth character; Seventh character SD645: First character of extension; 2Eh(.) SD646: Third character of extension; Second character of extension	S (Status change)	New

Special Register List (Continued)

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9□□□
SD647	File register capacity	File register capacity	• Stores the data capacity of the currently selected file register in 1 K word units.	S (Status change)	New
SD648	File register block number	File register block number	• Stores the currently selected file register block number.	S (Status change)	D9035
SD650	Comment drive	Comment drive number	• Stores the comment drive number selected at the parameters or by the QCDSET instruction.	S (Status change)	New
SD651 SD652 SD653 SD654 SD655 SD656	Comment file name	Comment file name	• Stores the comment file name (with extension) selected at the parameters or by the QCDSET instruction in ASCII code. B15 to B8 B7 to B0 SD651: Second character / First character SD652: Fourth character / Third character SD653: Sixth character / Fifth character SD654: Eighth character / Seventh character SD655: First character of extension / 2EH(.) SD656: Third character of extension / Second character of extension	S (Status change)	New
SD660	Boot operation designation file	Boot designation file drive number	• Stores the drive number where the boot designation file (*.QBT) is being stored.	S (Initial)	New
SD661 SD662 SD663 SD664 SD665 SD666		File name of boot designation file	• Stores the file name of the boot designation file (*.QBT). B15 to B8 B7 to B0 SD661: Second character / First character SD662: Fourth character / Third character SD663: Sixth character / Fifth character SD664: Eighth character / Seventh character SD665: First character of extension / 2EH(.) SD666: Third character of extension / Second character of extension	S (Initial)	New

(6) Instruction-Related Registers

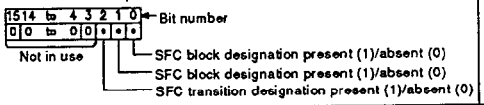
Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9□□□
SD714	Number of vacant communication request registration areas	0 to 32	• Stores the number of vacant blocks in the communications request area for remote terminal modules connected to the AJ71PT32-S3.	S (During execution)	M9081
SD715 SD716 SD717	IMASK instruction mask pattern	Mask pattern	• Patterns masked by use of the IMASK instruction are stored in the following manner: B15 to B0 SD715: b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 SD716: b31 b30 b29 b28 b27 b26 b25 b24 b23 b22 b21 b20 b19 b18 b17 b16 SD717: b47 b46 b45 b44 b43 b42 b41 b40 b39 b38 b37 b36 b35 b34 b33 b32	S (During execution)	New
SD718 SD719	Accumulator	Accumulator	• For use as replacement for accumulators used in A-series programs.	S/U	New
SD736	PKEY input	PKEY input	• SD that temporarily stores keyboard data input by means of the PKEY instruction.	S (During execution)	New

Special Register List (Continued)

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by
(When Set)</th><th>Corresponding
ACPU
D9□□□</th></tr>
<tr><td>SD738
SD739
SD740
SD741
SD742
SD743
SD744
SD745
SD746
SD747
SD748
SD749
SD750
SD751
SD752
SD753
SD754
SD755
SD756
SD757
SD758
SD759
SD760
SD761
SD762
SD763
SD764
SD765
SD766
SD767
SD768
SD769</td><td>Message storage</td><td>Message storage</td><td>• Stores the message designated by the MSG instruction.
<table><tr><th></th><th>B15 to B8</th><th>B7 to B0</th></tr><tr><td>SD738</td><td>2nd character</td><td>1st character</td></tr><tr><td>SD739</td><td>4th character</td><td>3rd character</td></tr><tr><td>SD740</td><td>6th character</td><td>5th character</td></tr><tr><td>SD741</td><td>8th character</td><td>7th character</td></tr><tr><td>SD742</td><td>10th character</td><td>9th character</td></tr><tr><td>SD743</td><td>12th character</td><td>11th character</td></tr><tr><td>SD744</td><td>14th character</td><td>13th character</td></tr><tr><td>SD745</td><td>16th character</td><td>15th character</td></tr><tr><td>SD746</td><td>18th character</td><td>17th character</td></tr><tr><td>SD747</td><td>20th character</td><td>19th character</td></tr><tr><td>SD748</td><td>22nd character</td><td>21st character</td></tr><tr><td>SD749</td><td>24th character</td><td>23rd character</td></tr><tr><td>SD750</td><td>26th character</td><td>25th character</td></tr><tr><td>SD751</td><td>28th character</td><td>27th character</td></tr><tr><td>SD752</td><td>30th character</td><td>29th character</td></tr><tr><td>SD753</td><td>32nd character</td><td>31st character</td></tr><tr><td>SD754</td><td>34th character</td><td>33rd character</td></tr><tr><td>SD755</td><td>36th character</td><td>35th character</td></tr><tr><td>SD756</td><td>38th character</td><td>37th character</td></tr><tr><td>SD757</td><td>40th character</td><td>39th character</td></tr><tr><td>SD758</td><td>42nd character</td><td>41st character</td></tr><tr><td>SD759</td><td>44th character</td><td>43rd character</td></tr><tr><td>SD760</td><td>46th character</td><td>45th character</td></tr><tr><td>SD761</td><td>48th character</td><td>47th character</td></tr><tr><td>SD762</td><td>50th character</td><td>49th character</td></tr><tr><td>SD763</td><td>52nd character</td><td>51st character</td></tr><tr><td>SD764</td><td>54th character</td><td>53rd character</td></tr><tr><td>SD765</td><td>56th character</td><td>55th character</td></tr><tr><td>SD766</td><td>58th character</td><td>57th character</td></tr><tr><td>SD767</td><td>60th character</td><td>59th character</td></tr><tr><td>SD768</td><td>62nd character</td><td>61st character</td></tr><tr><td>SD769</td><td>64th character</td><td>63rd character</td></tr></table></td><td>S (During execution)</td><td>New</td></tr>
</table>

Special Register List

(7) Debug

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by
(When Set)</th><th>Corresponding
ACPU
D9□□□</th></tr>
<tr><td>SD806
SD807
SD808
SD809
SD810
SD811</td><td>Status latch file name</td><td>Status latch file name</td><td>• Stores file name (with extension) from point in time when status latch was conducted as ASCII code.
B15 to B8 B7 to B0
SD806: Second character | First character
SD807: Fourth character | Third character
SD808: Sixth character | Fifth character
SD809: Eighth character | Seventh character
SD810: First character of extension | 2EH(.)
SD811: Third character of extension | Second character of extension</td><td>S (During execution)</td><td>New</td></tr>
<tr><td>SD812
SD813
SD814
SD815
SD816</td><td>Status latch step</td><td>Status latch step</td><td>• Stores step number from point in time when status latch was conducted
SD812: Pattern*
SD813: Block No.
SD814: Step No./transition No.
SD815: Sequence step No. (L)
SD816: Sequence step No.(H)
* Contents of pattern data
 
</td><td>S (During execution)</td><td>D9055
format change</td></tr>
</table>

Special Register List

(8) Latch area

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by
(When Set)</th><th>Corresponding
ACPU
D9□□□</th></tr>
<tr><td>SD900</td><td>Drive where power was interrupted</td><td>Access file drive number during power loss</td><td>• Stores drive number if file was being accessed during power loss.</td><td>S (Status change)</td><td>New</td></tr>
<tr><td>SD901
SD902
SD903
SD904
SD905
SD906</td><td>File name active during power loss</td><td>Access file name during power loss</td><td>• Stores file name (with extension) in ASCII code if file was being accessed during power loss.
<table><tr><th></th><th>B15 to B8</th><th>B7 to B0</th></tr><tr><td>SD901</td><td>2nd character</td><td>1st character</td></tr><tr><td>SD902</td><td>4th character</td><td>3rd character</td></tr><tr><td>SD903</td><td>6th character</td><td>5th character</td></tr><tr><td>SD904</td><td>8th character</td><td>7th character</td></tr><tr><td>SD905</td><td>1st character of extension</td><td>2E_H(.)</td></tr><tr><td>SD906</td><td>3rd character of extension</td><td>2nd character of extension</td></tr></table></td><td>S (Status change)</td><td>New</td></tr>
<tr><td>SD910
SD911
SD912
SD913
SD914
SD915
SD916
SD917
SD918
SD919
SD920
SD921
SD922
SD923
SD924
SD925</td><td>RKEY input</td><td>RKEY input</td><td>• Stored in sequence that PU key code was entered.
<table><tr><th></th><th>B15 to B8</th><th>B7 to B0</th></tr><tr><td>SD910</td><td>2nd character</td><td>1st character</td></tr><tr><td>SD911</td><td>4th character</td><td>3rd character</td></tr><tr><td>SD912</td><td>6th character</td><td>5th character</td></tr><tr><td>SD913</td><td>8th character</td><td>7th character</td></tr><tr><td>SD914</td><td>10th character</td><td>9th character</td></tr><tr><td>SD915</td><td>12th character</td><td>11th character</td></tr><tr><td>SD916</td><td>14th character</td><td>13th character</td></tr><tr><td>SD917</td><td>16th character</td><td>15th character</td></tr><tr><td>SD918</td><td>18th character</td><td>17th character</td></tr><tr><td>SD919</td><td>20th character</td><td>19th character</td></tr><tr><td>SD920</td><td>22nd character</td><td>21st character</td></tr><tr><td>SD921</td><td>24th character</td><td>23rd character</td></tr><tr><td>SD922</td><td>26th character</td><td>25th character</td></tr><tr><td>SD923</td><td>28th character</td><td>27th character</td></tr><tr><td>SD924</td><td>30th character</td><td>29th character</td></tr><tr><td>SD925</td><td>32nd character</td><td>31st character</td></tr></table></td><td>S (During execution)</td><td>New</td></tr>
</table>

(9) A to QnA conversion correspondences

ACPU special registers D9000 to D9255 correspond to the special registers SD1000 to SD1255 after A-series to the QnA-series conversion. These special registers are all set by the system, and users cannot use them to set program data.

Users who need to set data with these registers should edit the special registers for the QnA.

However, before conversion users could set data at special registers D9200 to D9255 only, and after conversion users can also set data at registers 1200 to 1255.

For more detailed information concerning the contents of the ACPU special registers, see the individual CPU users manual, and the MELSECNET and MELSECNET/B data link system reference manual.

Special Register List

ACPU Special Register	Special Register after Conversion	Special Register for Modification	Name	Meaning
D9000	SD1000	SD60	Fuse blown	Number of module with blown fuse
D9002	SD1002	SD61	I/O module verification error	I/O module verification error module number
D9004	SD1004	SD54	MINI link errors	Stores setting status made at parameters (modules 1 to 8)
D9005	SD1005	SD53	AC DOWN counter	Number of times for AC DOWN
D9008	SD1008	SD0	Self-diagnostic error	Self-diagnostic error number
D9009	SD1009	SD62	Annunciator detection	F number at which external failure has occurred
D9010	SD1010	—	Error step	Step number at which operation error has occurred.
D9011	SD1011	—	Error step	Step number at which operation error has occurred.
D9014	SD1014	—	I/O control mode	I/O control mode number
D9015	SD1015	SD203	Operating state of CPU	Operating state of CPU
D9016	SD1016	—	Program number	Stores sequence program under execution as BIN value
D9017	SD1017	SD520	Scan time	Minimum scan time (10 ms units)
D9018	SD1018	SD524	Scan time	Scan time (10 ms units)
D9019	SD1019	SD526	Scan time	Maximum scan time (10 ms units)
D9020	SD1020	—	Constant scan	Constant scan time (User sets in 10 ms units)
D9021	SD1021	—	Scan time	Scan time (in 1 ms units)
D9022	SD1022	SD412	Time	Time
D9025	SD1025	SD210	Clock data	Clock data (year, month)
D9026	SD1026	SD211	Clock data	Clock data (day, hour)
D9027	SD1027	SD212	Clock data	Clock data (minute, second)
D9028	SD1028	SD213	Clock data	Clock data (day of week)

ACPU Special Register	Special Register after Conversion	Special Register for Modification	Name	Meaning
D9035	SD1035	SD648	Extension file register	Use block No.
D9036	SD1036		Extension file register for designation of device number	Device number when individual devices from extension file register are directly accessed
D9037	SD1037			
D9038	SD1038	SD207	LED display priority ranking	Priorities 1 to 4
D9039	SD1039	SD208		Priorities 5 to 7
D9044	SD1044	—	For sampling trace	Step or time during sampling trace
D9049	SD1049	—	Work area for SFC	Block number of extension file register
D9050	SD1050	—	SFC program error number	Error code generated by SFC program
D5051	SD1051	—	Error block	Block number where error occurred
D5052	SD1052	—	Error step	Step number where error occurred
D5053	SD1053	—	Error transition	Transition condition number where error occurred
D5054	SD1054	—	Error sequence step	Sequence step number where error occurred
D9055	SD1055	SD815	Status latch	Status latch step
D9072	SD1072	—	PC communications check	Computer link data check
D9081	SD1081	SD714	Number of empty blocks in communications request registrtion area	Number of empty blocks in communications request registration area
D9085	SD1085	—	Register for setting time check value	Default value 10s
D9090	SD1090	—	Number of special functions modules over	Number of special functions modules over
D9091	SD1091	—	Detailed error code	Self-diagnosis detailed error code
D9094	SD9094	SD251	Head I/O number for replacement	Head I/O number for replacement

Special Register List (Continued)

ACPU Special Register	Special Register after Conversion	Special Register for Modification	Name	Meaning
D9100	SD1100	SD1300	Fuse blown module	Bit pattern in units of 16 points, indicating the modules whose fuses have blown
D9101	SD1101	SD1301		
D9102	SD1102	SD1302		
D9103	SD1103	SD1303		
D9104	SD1104	SD1304		
D9105	SD1105	SD1305		
D9106	SD1106	SD1306		
D9107	SD1107	SD1307		
D9116	SD1116	SD1400	I/O module verification error	Bit pattern, in units of 16 points, indicating the modules with verification errors.
D9117	SD1117	SD1401		
D9118	SD1118	SD1402		
D9119	SD1119	SD1403		
D9120	SD1120	SD1404		
D9121	SD1121	SD1405		
D9122	SD1122	SD1406		
D9123	SD1123	SD1407		
D9124	SD1124	SD63	Annunciator detection quantity	Annunciator detection quantity
D9125	SD1125	SD64	Annunciator detection number	Annunciator detection number
D9126	SD1126	SD65		
D9127	SD1127	SD66		
D9128	SD1128	SD67		
D9129	SD1129	SD68		
D9130	SD1130	SD69		
D9131	SD1131	SD70		
D9132	SD1132	SD71		
D9200	SD1200	—	LRDP processing results	0 : Normal end 2 : LRDP instruction setting fault 3 : Error at relevant station 4 : Relevant station LRDP execution disabled
D9201	SD1201	—	LWTP processing results	0 : Normal end 2 : LWTP instruction setting fault 3 : Error at relevant station 4 : Relevant station LWTP execution disabled
D9202	SD1202	—	Local station link type	Stores conditions for up to numbers 1 to 16
D9203	SD1203	—		Stores conditions for up to numbers 17 to 32
D9241	SD1241	—		Stores conditions for up to numbers 33 to 48
D9242	SD1242	—		Stores conditions for up to numbers 49 to 64
D9204	SD1204	—	Link state	0 : Forward loop, during data link 1 : Reverse loop, during data link 2 : Loopback implemented in forward/reverse directions 3 : Loopback implemented only in forward direction 4 : Loopback implemented only inreverse direction 5 : Data link disabled

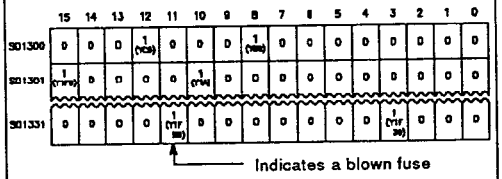
ACPU Special Register	Special Register after Conversion	Special Register for Modification	Name	Meaning
D9205	SD1205	—	Station implementing loopback	Station that implemented forward loopback
D9206	SD1206	—	Station implementing loopback	Station that implemented reverse loopback
D9207	SD1207	—	Link scan time	Maximum value
D9208	SD1208	—		Minimum value
D9209	SD1209	—		Present value
D9210	SD1210	—	Number of retries	Stored as cumulative value
D9211	SD1211	—	Number of times loop selected	Stored as cumulative value
D9212	SD1212	—	Local station operation state	Stores conditions for up to numbers 1 to 16
D9213	SD1213	—		Stores conditions for up to numbers 17 to 32
D9214	SD1214	—		Stores conditions for up to numbers 33 to 48
D9215	SD1215	—		Stores conditions for up to numbers 49 to 64
D9216	SD1216	—	Local station error detect state	Stores conditions for up to numbers 1 to 16
D9217	SD1217	—		Stores conditions for up to numbers 17 to 32
D9218	SD1218	—		Stores conditions for up to numbers 33 to 48
D9219	SD1219	—		Stores conditions for up to numbers 49 to 64
D9220	SD1220	—	Local station parameters non-conforming; remote I/O station I/O allocation error	Stores conditions for up to numbers 1 to 16
D9221	SD1221	—		Stores conditions for up to numbers 17 to 32
D9222	SD1222	—		Stores conditions for up to numbers 33 to 48
D9223	SD1223	—		Stores conditions for up to numbers 49 to 64
D9224	SD1224	—	Local station and remote I/O station initial communications underway	Stores conditions for up to numbers 1 to 16
D9225	SD1225	—		Stores conditions for up to numbers 17 to 32
D9226	SD1226	—		Stores conditions for up to numbers 33 to 48
D9227	SD1227	—		Stores conditions for up to numbers 49 to 64
D9228	SD1228	—	Local station and remote I/O station error	Stores conditions for up to numbers 1 to 16
D9229	SD1229	—		Stores conditions for up to numbers 17 to 32
D9230	SD1230	—		Stores conditions for up to numbers 33 to 48
D9231	SD1231	—		Stores conditions for up to numbers 49 to 64

Special Register List (Continued)

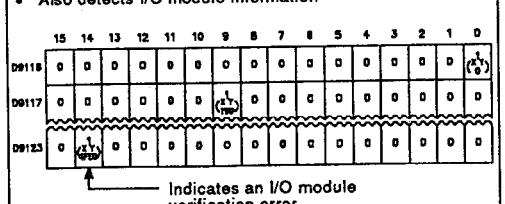
ACPU Special Register	Special Register after Conversion	Special Register for Modification	Name	Meaning
D9232	SD1232	—	Local station and remote I/O station loop error	Stores conditions for up to numbers 1 to 8
D9233	SD1233	—		Stores conditions for up to numbers 9 to 16
D9234	SD1234	—		Stores conditions for up to numbers 17 to 24
D9235	SD1235	—		Stores conditions for up to numbers 25 to 32
D9236	SD1236	—		Stores conditions for up to numbers 33 to 40
D9237	SD1237	—		Stores conditions for up to numbers 41 to 48
D9238	SD1238	—		Stores conditions for up to numbers 49 to 56
D9239	SD1239	—		Stores conditions for up to numbers 57 to 64
D9240	SD1240	—	Number of times communications errors detected	Stores cumulative total of receive errors
D9243	SD1243	—	Station number information for host station	Stores station number (0 to 64)

ACPU Special Register	Special Register after Conversion	Special Register for Modification	Name	Meaning
D9244	SD1244	—	Number of link device stations	Stores number of slave stations
D9245	SD1245	—	Number of times communications errors detected	Stores cumulative total of receive errors
D9248	SD1248	—	Local station operation state	Stores conditions for up to numbers 1 to 16
D9249	SD1249	—		Stores conditions for up to numbers 17 to 32
D9250	SD1250	—		Stores conditions for up to numbers 33 to 48
D9251	SD1251	—		Stores conditions for up to numbers 49 to 64
D9252	SD1252	—	Local station error conditions	Stores conditions for up to numbers 1 to 16
D9253	SD1253	—		Stores conditions for up to numbers 17 to 32
D9254	SD1254	—		Stores conditions for up to numbers 33 to 48
D9255	SD1255	—		Stores conditions for up to numbers 49 to 64

(10) Fuse blown module

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9□□□
SD1300	Fuse blown module	Bit pattern in units of 16 points, indicating the modules whose fuses have blown 0 : No blown fuse 1 : Blown fuse present	• The numbers of output modules whose fuses have blown are input as a bit pattern (in units of 16 points). (If the module numbers are set by parameter, the parameter-set numbers are stored.) • Also detects blown fuse condition at remote station output modules 	S (Error)	D9100
SD1301					D9101
SD1302					D9102
SD1303					D9103
SD1304					D9104
SD1305					D9105
SD1306					D9106
SD1307					D9107
SD1308					New
SD1309 to SD1330					New to New
SD1331					New

(11) I/O module verification

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding ACPU D9□□□
SD1400	I/O module verification error	Bit pattern, in units of 16 points, indicating the modules with verification errors. 0 : No I/O verification errors 1 : I/O verification error present	• When the power is turned on, the module numbers of the I/O modules whose information differs from the registered I/O module information are set in this register (in units of 16 points). (If the I/O numbers are set by parameter, the parameter-set numbers are stored.) • Also detects I/O module information 	S (Error)	D9116
SD1401					D9117
SD1402					D9118
SD1403					D9119
SD1404					D9120
SD1405					D9121
SD1406					D9122
SD1407					D9123
SD1408					New
SD1409 to SD1430					New to New
SD1431					New

APPENDIX 4 CAUTIONS WHEN ADAPTING AN EXISTING MELSEC-A SERIES PROGRAM FOR USE WITH QnACPU

To adapt a sequence program created for use with AnNCPU, AnACPU or AnUCPU is used for QnACPU, convert it using the A → QnA conversion option in the option menu of the file maintenance mode.
For details on GPPQ operations, refer to the SW□IVD-GPPQ GPP Function Operating Manual (Offline).
For details on instructions and devices, refer to the QnACPU Programming Manual (Common Instructions).
The instructions, devices, comments, etc., indicated below may require modification in the various modes after conversion.

4.1 Instructions

An□CPU Instruction	Instruction after A → QnA Conversion	How Dealt with
BMOVR instruction (Program example): LEDA BMOVR LEDC D10 LEDC D100 SUB K10 LEDR	 OUT SM1255 LEDC D10 LEDC D100 OUT SM1255 LEDR	Modified to BMOV instruction BMOV ZR100 ZR1000 K10
BXCHR instruction (Program example): LEDA BXCHR LEDC D10 LEDC D100 SUB K10 LEDR	 OUT SM1255 LEDC D10 LEDC D100 OUT SM1255 LEDR	Modified to BXCH instruction BXCH ZR100 ZR1000 K10
CHG instruction	OUT SM1255	Since QnACPU does not have the main/subsequence program system, there is no CHG instruction. OUT SM1255 is not necessary and therefore deleted. The main/subsequence program requires modification after conversion, and new parameter settings must be made. (See Appendix 4.5.)
CHK instruction (Program example): CHK M10 X100	 CHK	See Appendix 4.12.
CLC instruction (Program example): CLC	 RST SM1012	Modified to special relay for carry flag SM700 RST SM700
AnA/AnUCPU dedicated instructions IX instruction	 OUT SM1255	See Appendix 4.12.

AnCPU Instruction	Instruction after A → QnA Conversion	How Dealt with
LEDA instruction (excluding dedicated instructions for AnACPU, AnUCPU) (Program example): LEDA ABCDEFGH	OUT SM1255	Modified with the LED instruction $MOV "ABCDEFGH" D0 $MOV "IJKLMNOP" D10 $+ D0 D10 D20 LED D20 LED display is performed after adding the right 8 characters and the left 8 characters.
LEDB instruction (excluding AnACPU, AnUCPU dedicated instructions) (Program example): LEDB IJKLMNOP	OUT SM1255	
LRDP instruction (Program example:) LRDP K3 D10 D100 K10	OUT SM1255	Modified to ZNRD instruction J.ZNRD J0 K3 D10 D100 K10 M0
LWTP instruction (Program example): LWTP K3 D10 D100 K10	OUT SM1255	Modified to ZNWR instruction J.ZNWR J0 K3 D10 D100 K10 M0
OUT instruction (Program example): Setting the number of points and set value for a counter by parameter Number of points: 512 Device used for set value: D3000 OUT C0 K10 OUT C256 D3000	OUT C0 K10 OUT C256 D3000	After conversion, the parameters will be set to the defaults, so they must be set again if using an interrupt counter.
RFRP instruction (Program example): RFRP H100 K10 W100 K10	OUT SM1255	Modified to RFRP instruction for QnACPU U.RFRP U10 K10 W100 K10 M0
RTOP instruction (Program example): RTOP H100 K10 W100 K10	OUT SM1255	Modified to RTOP instruction for QnACPU U.RTOP U10 K10 W100 K10 M0
SCMP instruction (Program example): LEDA SCAP LEDC D10 LEDC D100 LEDC M0 LEDR	OUT SM1255 LEDC D10 LEDC D100 LEDC M0	Modified to AND$= instruction and OUT instruction AND$= D0 D100 OUT M0
SEG instruction (When used as a partial refresh instruction) (Program example:) SET M9052 SEG K4Y10 K4B1	SET M1052 SEG K4Y10 K4B1	Modified to RFS instruction RFS Y10 H8
STC instruction (Program example): STC	SET M1012	Modified to special relay for carry flag SM700 SET SM700

AnCPU Instruction	Instruction after A → QnA Conversion	How Dealt with
SUB instruction	OUT SM1255	Since QnACPU cannot store microcomputer programs, there is no SUB instruction. OUT SM1255 is not necessary and therefore deleted. AnNCPU and A3HCPU microcomputer programs are converted to sequence programs using QnACPU instructions. (See Appendix 4.6.)
ZRRD instruction (Program example): DMOV K8000 D9036 LEDA ZRRD	DMOV K8000 SD1036 OUT SM1255	Modified to MOV instruction MOV SD718 ZR8000 SD718 is the device resulting from converting accumulator A0.
ZRWR instruction (Program example): DMOV K8000 D9036 LEDA ZRWR	DMOV K8000 SD1036 OUT SM1255	Modified to MOV instruction MOV SD718 ZR8000 SD718 is the device resulting from converting accumulator A0.

(b) Instructions for which program modification is unnecessary after conversion

AnCPU Instruction	Instruction after A → QnA Conversion
ASC instruction (Program example): ASC ABCDEFGH D10	$MOV ABCDEFGH D10 Note: Since the $MOV instruction has 00_H appended at the end, 5 data register words (for 9 characters) must be secured.
DFLOAT instruction (Program example): LEDA DFLOAT LEDC D10 LEDC D100 LEDR	DFLT D10 D100
DOUT instruction (Program example): LEDA DOUT LEDC Y10 LEDR	OUT DY10
DRCL instruction (Program example): DRCL K8	DRCL SD718 K8 SD718 is the device resulting from converting accumulator A0.
DRCR instruction (Program example): DRCR K8	DRCR SD718 K8 SD718 is the device resulting from converting accumulator A0.

AnCPU Instruction	Instruction after A → QnA Conversion
DROL instruction (Program example): DROL K8	DROL SD718 K8 SD718 is the device resulting from converting accumulator A0.
DROR instruction (Program example): DROR K8	DROR SD718 K8 SD718 is the device resulting from converting accumulator A0.
DRST instruction (Program example): LEDA DRST LEDC Y10 LEDR	RST DY10
DSUM instruction (Program example): DSUM D10	DSUM D10 SD718 SD718 is the device resulting from converting accumulator A0.
DSET instruction (Program example): LEDA DSET LEDC Y10 LEDR	SET DY10
FLOAT instruction (Program example): LEDA FLOAT LEDC D10 LEDC D100 LEDR	FLT D10 D100
OUT instruction (Program example): First number in parameters Low speed : 0 High speed: 200 Retentive : 224 Extension timer Low speed : 256 High speed: 512 Retentive : 768 Device used for set value: Set D5000 OUT T0 K10 OUT T200 K10 OUT T225 K10 OUT T256 D5000 OUT T512 D5256 OUT T768 D5512	OUT T0 K10 OUTH T200 K10 OUT ST225 K10 OUT T256 D5000 OUTH T512 D5256 OUT ST768 D5512
RCL instruction (Program example): RCL K8	RCL SD718 K8 SD718 is the device resulting from converting accumulator A0.

AnCPU Instruction	Instruction after A → QnA Conversion
RCR instruction (Program example): RCR K8	RCR SD718 K8 SD718 is the device resulting from converting accumulator A0.
ROL instruction (Program example): ROL K8	ROL SD718 K8 SD718 is the device resulting from converting accumulator A0.
ROR instruction (Program example): ROR K8	ROR SD718 K8 SD718 is the device resulting from converting accumulator A0.
SADD instruction (Program example): LEDA SADD LEDC D10 LEDC D100 LEDC D200 LEDR	$+ D10 D100 D200
SER instruction (Program example): SER D10 D100 K5	SER D10 D100 SD718 K5 SD718 is the device resulting from converting accumulator A0.
SMOV instruction (Program example): LEDA SMOV LEDC D10 LEDC D100 LEDR	$MOV D10 D100
SUM instruction (Program example): SUM D10	SUM D10 SD718 SD718 is the device resulting from converting accumulator A0.
ZRRDB instruction (Program example): DMOV K8000 D9036 LEDA ZRRDB	DMOV K8000 SD1036 ZRRDB SD1036 SD718 SD1036 is the device resulting from converting special register D9036. SD718 is the device resulting from converting accumulator A0.
ZRWRB instruction (Program example): DMOV K8000 D9036 LEDA ZRWRB	DMOV K8000 SD1036 ZRWRB SD1036 SD718 SD1036 is the device resulting from converting special register D9036. SD718 is the device resulting from converting accumulator A0.

AnCPU Instruction	Instruction after A → QnA Conversion
AnA/AnUCPU dedicated instructions LEDA/LEDB instruction name SUB/LEDC device 1 ..... SUB/LEDC device n LEDR (Program example 1: SIN instruction): LEDA SIN LEDC D10 LEDC D100 LEDR (Program example 2: DSER instruction): LEDA DSER LEDC D10 LEDC D100 SUB K5 LEDR	Instruction name device 1 .... device n SIN D10 D100 DSER D10 D100 SD718 K5 SD718 is the device resulting from converting accumulator A0.
AnA/AnUCPU special function module dedicated instructions LEDA/LEDB instruction name SUB/LEDC device 1 ..... SUB/LEDC device n LEDR (Program example): LEDA SVWR1 SUB H2 LEDC D10 LEDR	Enter G. before the instruction. G. instruction name device Un ···· device n G.SVWR1 U2 D10
AnA/AnUCPU data link dedicated instructions LEDA/LEDB instruction name SUB/LEDC device 1 ..... SUB/LEDC device n LEDR (Program example): LEDA LRDP SUB K12 LEDC D10 LEDC D100 SUB K5 LEDC M0 LEDR	Enter J. before the instruction. J. instruction name J0 device 1 ···· device n Network N. when using MELSECNET II J.ZNRD J0 K12 D10 D100 K5 M0

4.2 Devices

(a) Only devices within the QnACPU range are converted.

AnCPU Device		Device after A → QnA Conversion
X□□□		Same as to left
Y□□□		Same as to left
M□□□	M/L/S depend on the parameter settings.	Same as to left
L□□□		Same as to left
S□□□		Same as to left (correct to M□□□)
M9000 to M9255		SM1000 to SM1255
B□□□		Same as to left
T (low-speed timer)	Low speed/high speed/retentive determined by parameter setting.	Same as to left
T (high-speed timer)		Same as to left
T (retentive timer)		ST□□□
C□□□		Same as to left
F□□□		Same as to left
D□□□		Same as to left
D9000 to D9255		SD1000 to SD1255
W□□□		Same as to left
R□□□		Same as to left
Z□		Z → Z0 Z1 to Z6 → Z1 to Z6
V□		V → V7 V1 to V6 → V8 to V13
A0, A1		SD718, SD719
P□□□		Same as to left
I□□		Same as to left
N□		Same as to left
K□□□		Same as to left
H□□□		Same as to left

(b) Devices that are outside the QnACPU range are converted to SM1255 if they are bit devices and to SD1255 if they are word devices.

4.3 Parameters

The following parameter settings only are converted to QnACPU use.

- Latch range setting
  Converted to "latch clear key valid" range.
  The latch clear key invalid range is made blank (no setting).
- MELSECNET (II, /10) setting
  For the MELSECNET setting when the ACPU is an AnN or AnA, the number of modules aversion is stored, but the network refresh parameters are not converted.
- I/O allocation
  Only the head I/O No. is made blank; all other items are converted.
- MELSECNET/MINI automatic refresh setting
  If only I/O allocation was set in the parameters and MELSECNET/MINI automatic refresh settings have not been made, the MELSECNET/MINI data link operates with the default values.

The following items are set to the QnACPU default. If settings had been made, they must be made again.

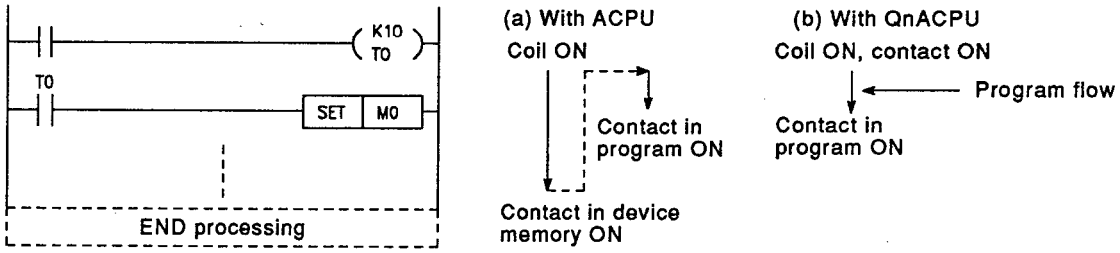
- RUN-PAUSE contact
- STOP → RUN output mode
- Interrupt counter No.
- WDT setting
- Operation mode at error

4.4 Operation of Timers and Interrupt Counters

(1) Timers

(a) The ACPU turns timer coils ON/OFF on execution of the OUT instruction, and updates timer present values and turns contacts ON/OFF on execution of the END instruction. In contrast, the QnACPU turns timer coils ON/OFF, updates present values, and turns contacts ON/OFF, on execution of the OUT instruction. Note that after conversion, the turning of contacts ON/OFF may be up to one scan faster.

Example: Timing for turning contact ON



In the case of ACPU, a timer contact will turn ON quickly if it is located in the first step. In the case of QnACPU, it will turn ON quickly if it is located in the step following OUT T.

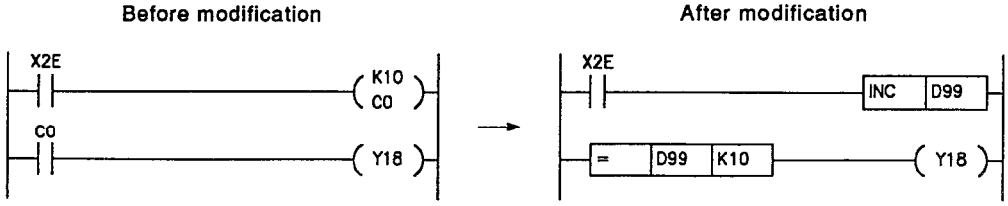
(b) Note that processing differs as follows when the set value of a timer is set as K0:

- For ACPU, count is in infinite units (timer does not count up).
- For QnACPU, the timer counts up instantaneously.

(2) Interrupt counter

QnACPU interrupt counters count the number of times an interrupt occurs. However, the counter contact does not turn ON even when the count has reached the set value.
The operation of ACPU interrupt counters differs according to the CPU type.

(a) A3HCPU, AnACPU and AnUCPU interrupt counters count the number of times that interrupts occur. However, when the count reaches the set value the contact is turned ON.
In order to achieve the same operation as with A3HCPU, AnACPU and AnUCPU interrupt counters when using a QnACPU, the program must be modified after conversion.
An example modification is shown below.



(b) AnCPU and AnNCPU interrupt counters are used in interrupt programs.
In order to achieve the same operation as with AnCPU and AnNCPU interrupt counters when using a QnACPU, the program must be modified after conversion.
When ordinary counters are used in an interrupt program with QnACPU, they operate in the same way as with AnNCPU.

4.5 Sequence Programs, Statements, Notes

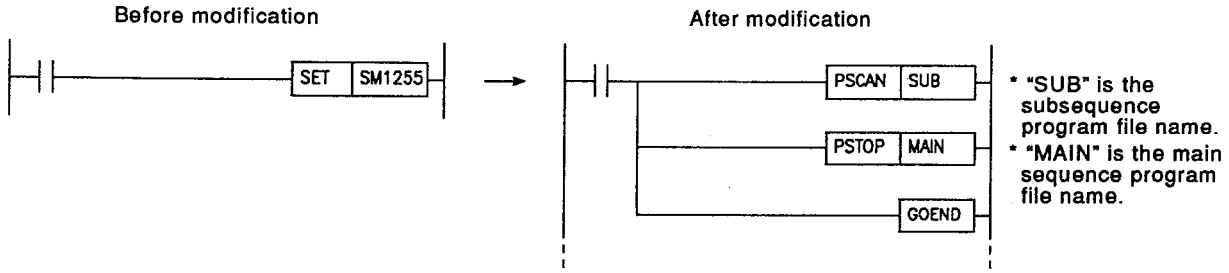
After conversion by A → QnA conversion, sequence programs are stored in the set file.
If there is a subsequence program, the main/subsequence program must be modified.
There are two types of modification, as indicated below:

(a) When the main sequence program and subsequence program are executed alternately, the parameters and program are modified according to the following procedure.

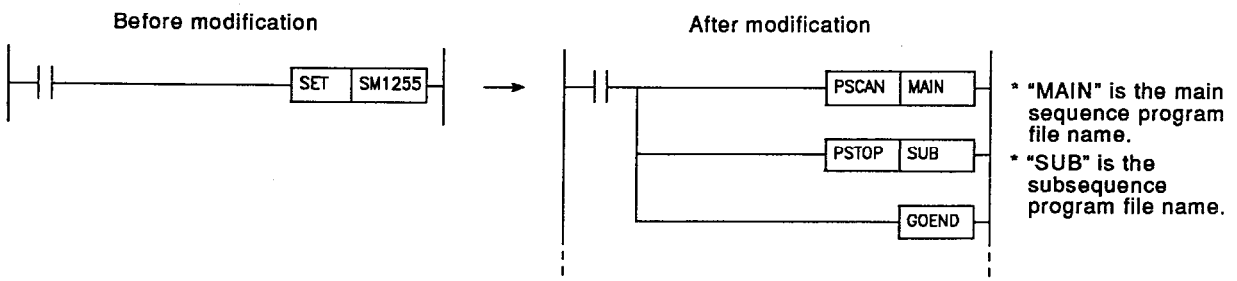
1) Modification of parameters
Set the file names of the main sequence program and subsequence program in program setting in the auxiliary settings in the parameter mode. The execution type can be selected as scan execution or standby execution for the main sequence program.

2) Modification of the sequence program

- The CHG instruction that switches between main sequence and subsequence programs is converted to OUT SM1255 in A → QnA conversion. Modify this OUT SM1255 to the PSCAN instruction which converts another sequence program to scan execution.
- Next, add the GOEND instruction that executes a jump to the END instruction, and the next step.
- Next, add the PSTOP instruction, which converts another sequence program to standby execution, to the first step of the sequence program.
This makes execution from the main sequence program to the subsequence program possible, and makes it possible not to execute a subsequence program without executing the main sequence program.
- Main sequence program



- Subsequence program



(b) When the main sequence program and subsequence program are executed serially as one program, the parameters and program are modified according to the following procedure.

1) Modification of parameters

Set the file names in the order main sequence program then subsequence program in program setting in the auxiliary settings in the parameter mode.
Select scan execution as the execution type can for both the main sequence program and the subsequence program.

2) Modification of the sequence program

- The CHG instruction that switches between main sequence and subsequence programs is converted to OUT SM1255 in A → QnA conversion. Since this is not required with QnACPU, delete it.
- If the same interrupt program or pointer is used for the main sequence program and subsequence program, use only one interrupt program or pointer.

REMARK

An ACPU executes END processing on switching from execution of the main sequence program to execution of the subsequence program, and also executes END processing after execution of the subsequence program.
When a QnACPU executes the two programs consecutively, note that END processing is only executed after execution of the second program.

Statements and notes are entered in the sequence program file after A → QnA conversion. No modification is required after conversion.

4.6 Microcomputer Programs

Microcomputer programs and utility software packages cannot be converted because QnACPU has no microcomputer mode.
When microcomputer programs and utility software packages are used with ACPU, a SUB instruction (microcomputer program call instruction) is written in the sequence program to execute them. After A → QnA conversion, the SUB instruction is converted to OUT SM1255, but this is not necessary and should be deleted.
In the case of user-created microcomputer programs, convert the processing contents of the microcomputer program to a sequence program using the additional operation instructions featured with QnACPU.
When using the utility software packages listed below, convert the processing contents of the utility software package to a sequence program using operation instructions added with QnACPU.

- SW□□□-AD57P ..... See the QnACPU Programming Manual (AD57 Instructions)
- SW□□□-UTLP-FN0 ... See the QnACPU Programming Manual (Common Instructions)
- SW□□□-UTLP-FN1 ... See the QnACPU Programming Manual (Common Instructions)
- SW□□□-UTLP-PID ... See the QnACPU Programming Manual (PID Control Instructions)

4.7 Comments

Conversions are only made for the devices within the QnACPU range.
Devices outside the QnACPU range are not converted.

4.8 Constant Scan Function, Error Check Function

When the constant scan function and error check function are used with ACPU, special registers and special relays are set.
In contrast, with QnACPU, these functions are set in the parameters. In order to use these functions after conversion, they must be set in PC RAS setting in the parameter mode.

4.9 I/O Control Mode

The I/O control mode for QnACPU is the refresh mode (direct I/O by means of devices is also possible).

- Since the I/O control mode for AnACPU and AnUCPU is also the refresh mode, there are no problems with the input timing of inputs (X) or the output timing of outputs (Y).
- However, in the case of AnCPU, AnNCPU, and A3HCPU, the I/O control mode is either fixed as the direct mode or selectable, and the input timing for inputs (X) and output timing for outputs (Y) therefore differs from that for the refresh mode.
  The I/O control mode for each CPU module is shown below.

Model	Control Mode
AnCPU	Fixed as the direct mode
AnNCPU	Direct mode or refresh mode selected with switch on module
A3HCPU	Direct mode or refresh mode selected by parameter setting
AnACPU	Fixed as the refresh mode
AnUCPU	Fixed as the refresh mode

- Modifying programs that generate pulses from SET/RST instructions, by using direct devices
  Programs which, in the direct mode, output pulse output to external destinations using SET/RST instructions, are modified to programs that use direct output devices for QnACPU.

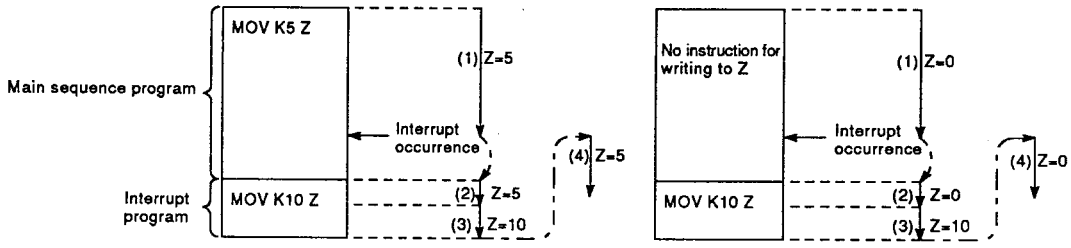
4.10 Data Link Systems

- AnUCPU data link systems
  The network settings in the AnUCPU parameters can be converted by A → QnA conversion. The parameters do not need to be modified after conversion.
- CPUs other than AnUCPU
  The link settings in the CPU parameters cannot be converted in A → QnA conversion. Link settings must be made in the parameters after conversion.

4.11 Index Register Processing

With QnACPU, the contents of index registers change when program processing transfers between the main sequence program and interrupt programs.

- Transfer of program processing from main sequence program to interrupt program
  The contents of the index registers of the main sequence program are saved, and then these contents are passed to the interrupt program.
- Transfer of program processing from interrupt program to main sequence program
  The index registers in the interrupt program are cleared, and the saved main sequence program contents are written to them.



With ACPU, processing differs according to the CPU type.

- The processing for AnACPU and AnUCPU is the same as for QnACPU, and therefore no program modification is required after conversion.
- In the case of AnCPU, AnNCPU and A3HCPU, when program processing is transferred from an interrupt program to the main program, the data updated in the interrupt program are passed on to the main program.
  If, for example, the value written to an index register in an interrupt program is passed on to the main sequence program, modify the program so that the value is passed on via a data register.

4.12 CHK Instruction, IX Instruction

(1) CHK instruction

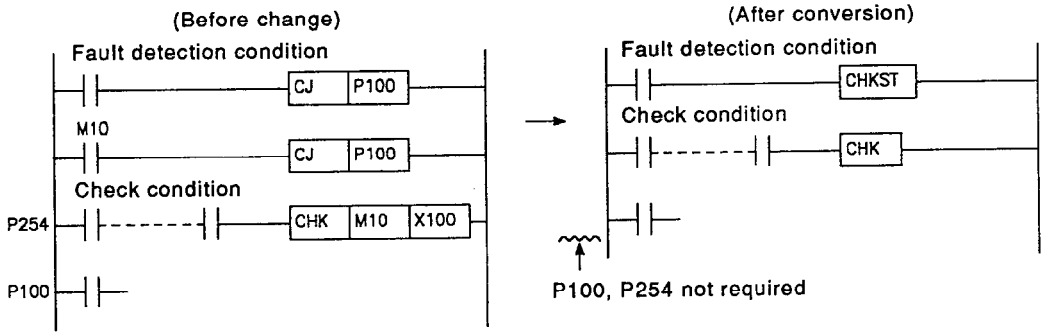
The CHK instruction operates as a fault check instruction in the case of the QnACPU.
With the ACPU, there are two types of processing according to the CPU type.

(a) Fault check..... AnCPU, AnNCPU (direct I/O control mode), A3HCPU, AnACPU, AnUCPU

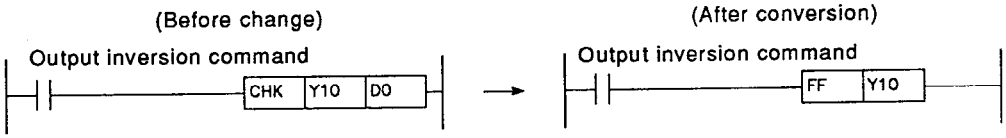
(b) Bit device output inversion.... AnNCPU (refresh I/O control mode)

After conversion, program modification is required for each processing.

(a) For fault check
Change the CJ instruction in the step before the CHK instruction to a CHKST instruction.



(b) For bit device output inversion
QnACPU has the FF instruction for inverting bit device outputs.
Change the CHK instruction to the FF instruction.



(2) IX command

The IX instruction is converted, but not executed. Modify the program so that all the devices that are objects of the IX instruction are subject to index qualification.

4.13 Accessing File Registers, R, with Instructions

With QnACPU, no error will occur even if an attempt is made to execute instructions that access file registers outside the setting range.
When reading data, $FFFF_H$ is stored in the storage destination device.
When writing data, the instruction is executed but no data is stored in the file register.
With ACPU, execution of such an instruction causes an error.

The capacities of file registers, R, are set by parameter. It is therefore necessary to check the capacities of file registers before executing instructions that access file registers, such as the MOV instruction and + instruction.

APPENDIX 5 ERROR CODES RETURNED TO THE REQUEST SOURCE IN GENERAL DATA PROCESSING

With QnACPU, when an error occurs as a result of a request for general data processing from a peripheral device, special function module, or network system, the error code is returned to the source of the general data processing request.

POINT

Since this error code is not an error detected by the QnACPU self-diagnosis function, it is not stored in special relay SD0.
If the request source is a peripheral device, it displays a message or the error code.
If the request source is a special function module or network system, the error code corresponding to the requested processing is returned.

5.1 Error Codes

The error numbers of error codes differ according to the location where the error was detected.
The correspondences between locations where errors are detected and error codes are indicated in the table below.

Location where Error Detected	Error Code (Hexadecimal)	Reference Destination for Error Contents
CPU	4000_H to $4FFF_H$	See Appendix 5.2.
Special function module	7000_H to $7FFF_H$	User's manual for each special function module
Network system	$F000_H$ to $FFFF_H$	MELSECNET/10 Network System Reference Manual for QnACPU

5.2 Error Contents of Error Codes Detected by the CPU (4000H to 4FFFH)

The error contents of error codes detected by the CPU (4000H to 4FFFH) and the message displayed at the peripheral device on their detection, are indicated in the table below.

Error Code (Hexadecimal)	Error	Error Details	Message Displayed at Peripheral Device	Corrective Action
4000H	CPU-related errors	Sum check error.	Message (1) is displayed.	Check the connection between the CPU and connecting cable.
4001H		Remote request that cannot be handled executed.	Message (1) is displayed.	Check the requested remote operation.
4002H				
4003H		Command for which global request is not possible executed.	Message (1) is displayed.	Check the requested command.
4004H		Since the QnACPU is system protected, the request contents cannot be executed.	Execution is not allowed during system protection.	Turn the QnACPU system protect switch OFF.
4005H		The data volume is too large for the designated request.	Cannot execute in excess of capacity.	Reduce the volume of data so that it can be handled with the request.
4006H		Password has not been cancelled.	Password has not been canceled.	Cancel the set password.
4007H		CID is different from QnACPU data.	Message (1) is displayed.	Check the CID.
4008H		QnACPU is not BUSY. (Buffer is not empty.)	Message (1) is displayed.	Re-execute the request after elapse of an arbitrary time period.
4010H	CPU mode error	The request contents cannot be executed because the QnACPU is in the RUN status.	Cannot execute when PC is in RUN mode.	Execute the request after setting the QnACPU to the STOP status.
4011H		The request contents cannot be executed because the QnACPU is not in the STEP RUN status.	Cannot execute while PC is not in STEP RUN mode.	Execute the request after setting the QnACPU to the STEP RUN status.
4021H	CPU file-related errors	Designated drive memory does not exist or is abnormal.	The target dirve contains a fault.	Check status of designated drive memory.
4022H		File with designated file name, file No. does not exist.	The file name does not exist.	Check the designated file name and file No..
4023H		File name and file No. of designated file do not match.	Cannot access files.	Delete the file and create a new one.
4024H		Designated file cannot be accessed by the user.	This file cannot be handled.	Do not access the designated file.
4025H		The designated file is processing a request from another source.	Alert (2) is displayed.	Forcibly execute the request. Or, execute the request again after the other processing has finished.
4026H		Password set for target drive memory must be designated.	Keyword doesn't match.	Access by designating the password set for the target drive memory.
4027H		The designated range exceeds the file range.	File capacity is not enough.	Check the designated range, and access within the permissible range.
4028H		The same file already exists.	Alert (3) is displayed.	Forcibly execute the request. Or, change the file name and then execute the request.

Error Code (Hexadecimal)	Error	Error Details	Message Displayed at Peripheral Device	Corrective Action
4029H	CPU file-related errors	The capacity of the designated file is not secured.	File capacity is not enough.	Review the capacity of the designated file. Or, sort the data in the designated drive memory and re-execute.
402AH		Designated cluster number does not exist.	Cannot access files.	Check the designated cluster No., and access by designating a cluster No. within the number of clusters of the designated drive memory.
402BH		The request contents cannot be executed with the designated drive memory.	Cannot access files.	Do not make requests which cause errors with the designated drive memory.
402CH		Currently the request contents cannot be executed.	Cannot access files.	Re-execute after elapse of an arbitrary time period.
4030H	CPU device designation error	Designated device name cannot be handled.	Device is invalid.	Check the designated device name.
4031H		Out-of-range device No. designated.	Device No. is out of range.	Check the designated device No..
4032H		Mistake in designated device qualification.	Device is invalid.	Check the method for qualification of the designated device.
4033H		The designated device is for system use and cannot be written to.	Device is invalid.	Do not write data to the designated device, or turn it ON/OFF.
4040H	Special function module designation error	The designated special function module cannot carry out the request.	The unit does not exist.	Do not issue requests that cause errors to the designated special function module.
4041H		Buffer memory range of special function module for which access range designated exceeded.	The # of devices is too large.	Check the first address and number of accessed points, and access within the actual ranges at the special function module.
4042H		Access to the designated special function module is not possible.	The corresponding unit is faulty.	Check if the designated special function module is operating normally.
4043H		The special function module is not at the designated position.	The unit does not exist.	Check the head I/O No. of the designated special function module.
4044H		A control bus error has occurred.	The corresponding unit is faulty.	Check if there is a fault in the hardware of the special function module or other modules.
4045H		Setting required for simulation have not been made.	Data error.	Make settings for the simulation.
4046H		The first number and number of points of the device designated for simulation is not in 16-point units.	Device No. is not in 16 units.	Check the first number and number of points of the device and modify to 16-point units.
4050H	Protect error	Request contents cannot be executed because the write protect switch of the memory card is ON.	Cannot execute as the memory protect switch is ON.	Set the write protect switch of the memory card to OFF.
4051H		Designated device memory cannot be accessed.	Wrong ROM	Check the following and take appropriate action. • Is the memory usable? • Is the designated drive memory installed correctly?

Error Code (Hexadecimal)	Error	Error Details	Message Displayed at Peripheral Device	Corrective Action
4052H	Protect error	Data cannot be written to the designated file because its attribute is "read only".	Write is prohibited.	Do not write data to the designated file. Or, change the file attribute.
4053H		Error occurred on writing data to the designated drive memory.	Cannot write correctly in ROM.	Check the designated drive memory. Or, replace the object drive memory then attempt writing again.
4054H		Error occurred on deleting data from the drive memory.	Cannot erase ROM correctly.	Check the designated drive memory. Or, replace the object drive memory then attempt deletion again.
4060H	Online registration errors	The CPU system area for registering monitor conditions is already being used by another device.	Alert (2) is displayed.	On completion of monitoring for the other device, re-attempt monitoring. Or, increase the system area of the internal memory by using a format with an option appended.
4061H		Communications failed.	Not registered.	Re-attempt communications.
4062H		Another device is already using the detailed condition for monitoring.	Alert (2) is displayed.	Do not use the detailed condition for monitoring from the designated device. Or, cancel the other device's monitoring detailed condition and re-attempt monitoring.
4063H		The number of registrations for file lock is greater than 16.	Cannot access files.	Reduce the number of registrations to 16 or less.
4064H		Incorrect setting contents.	Unable to execute due to on going process.	Correct the setting contents.
4065H		Device I/O parameter information differs from parameters.	Does not match the parameter.	Check the parameters. Or, check the data.
4066H		An entry that differs from the one set for the designated device memory was designated.	Keyword doesn't match.	Check the designated password.
4067H		The designated monitor file has not been secured.	File capacity is not enough.	Secure the monitor file, then perform monitoring.
4068H		The designated command cannot be registered or cancelled because it is being executed.	Unable to execute due to on going process.	Re-execute the command after requests from other devices are completed.
4069H		Condition already satisfied at device.	Setting is incorrect.	Check the monitor condition. Or, carry out monitor registering again and then execute monitoring.
406AH		Drive other than 1 to 3 has been designated.	Drive specification is incorrect.	Check the designated drive and designate a correct drive.
4070H	Ladder verification	The program before modification differs from the registered program.	Program does not match.	Check the registered program and match the program to it.

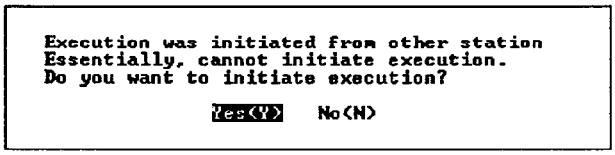
Error Code (Hexadecimal)	Error	Error Details	Message Displayed at Peripheral Device	Corrective Action
4080H	Other errors	Data error.	Data is faulty.	Check the requested data contents.
4081H		The searched object cannot be detected.	Cannot find the find target.	Check the data to be searched.
4082H		The designated command cannot be executed because it is already being executed.	Unable to execute due to on going process.	Execute the command when the request from the other device is completed.
4083H	Other errors	An attempt was made to execute a program not registered in the parameters.	Not registered.	Register the program to be executed in the parameters.
4084H		The designated pointer P, I cannot be detected.	Cannot find the find target.	Check the data to be searched.
4085H		Pointer P, I designation is not possible because the program is not registered in the parameters.	Not registered.	Registering the program to be executed in the parameters, then designate the pointer P, I.
4086H		An attempt has been made to add a pointer P, I that already exists.	Device ranges are duplicated.	Check the pointer No. to be added and change it.
4087H		The number of pointers designated is too great.	No pointer exists.	Check and correct the pointer designation.
4088H		The designated step No. is not at the head of the instruction.	Execution position is incorrect.	Check and correct the designated step No..
4089H		An END instruction was inserted/deleted while the CPU was in the RUN state.	Setting is incorrect.	Set the CPU to STOP before making the insertion/deletion.
408AH		The file capacity has been exceeded by write during RUN.	File capacity is not enough.	Set the CPU to STOP and then write the program.
408BH		Cannot execute remote request.	Data error.	Set the CPU to a state in which it can execute a remote request, then reissue the request.
4090H	Online registration error during SFC STEP RUN	Too many block break points.	Setting is out of range.	Check and correct the set number.
4091H		The number of registered block break points is incorrect.	Setting is out of range.	Check and correct the registered number.
4092H		Too many step break points.	Setting is out of range.	Check and correct the set number.
4093H		The number of registered step break points is incorrect.	Setting is out of range.	Check and correct the registered number.
4094H		An attempt was made to execute a request during block continuous processing.	Unable to execute due to on going process.	Reissue the request after processing is completed.
4095H		An attempt was made to execute a request during block forced execution processing.	Unable to execute due to on going process.	Reissue the request after processing is completed.
4096H		An attempt was made to execute a request during step continuous processing.	Unable to execute due to on going process.	Reissue the request after processing is completed.
4097H		An attempt was made to execute a request during step forced execution processing.	Unable to execute due to on going process.	Reissue the request after processing is completed.
4098H		An attempt was made to execute a request during 1 step continuous processing.	Unable to execute due to on going process.	Reissue the request after processing is completed.
4099H		An attempt was made to execute a request during 1 step forced execution processing.	Unable to execute due to on going process.	Reissue the request after processing is completed.

Error Code (Hexadecimal)	Error	Error Details	Message Displayed at Peripheral Device	Corrective Action
409AH	Online registration error during SFC STEP RUN	An attempt was made to execute a request during block forced end processing.	Unable to execute due to on going process.	Reissue the request after processing is completed.
409BH		An attempt was made to execute a request during step forced end processing.	Unable to execute due to on going process.	Reissue the request after processing is completed.
409CH	Online registration error during SFC STEP RUN	An attempt was made to execute a request during holding step reset processing.	Unable to execute due to on going process.	Reissue the request after processing is completed.
409DH		A block No. with no created block or out-of-range block No. has been designated.	Setting is incorrect.	Check and correct the setting.
409EH		A step No. for which no step has been created has been designated.	Setting is incorrect.	Check and correct the setting.
409FH		Out-of-range number of cycles designated.	Setting is out of range.	Check and correct the set number.
40A0H	SFC device designation error	Out-of-range block No. designated.	Setting is incorrect.	Check and correct the setting.
40A1H		Designation exceeds the range for the number of blocks.	Setting is out of range.	Check and correct the set number.
40A2H		Out-of-range step No. designated.	Setting is incorrect.	Check and correct the setting.
40A3H		Designation exceeds range for number of steps.	Setting is out of range.	Check and correct the set number.
40A4H		Out-of-range sequence step No. designated.	Setting is incorrect.	Check and correct the setting.
40A5H		Designated device was out-of-range.	Setting is out of range.	Check and correct the number of settings.
40A6H		The block designation pattern or step designation pattern was incorrect.	Setting is incorrect.	Check and correct the setting.
40B0H	SFC file-related errors	Designated drive is incorrect.	Setting is incorrect.	Check and correct the setting.
40B1H		Designated program does not exist.	The file name dose not exist.	Check and correct the designated file name.
40B2H		The designated program was not an SFC program.	This file cannot be handled.	Check and correct the designated file name.
40B3H		There was an SFC dedicated instruction in the write during RUN area.	Setting is incorrect.	Check and correct the setting.
4A00H	Link-related errors	Designated station cannot be accessed because no routing parameters have been set at the relevant station.	Routing parameter does not exist.	Set the routing parameters for accessing the designated station in the relevant station.
4A01H		No network with the No. set in the routing parameters exists.	The network I/O does not exist.	Check and correct the routing parameters set at the relevant station.
4A02H		Cannot access the designated station.	Link unit error.	Check if an error has occurred at the network module/link module, or if the online state has not been established.
4B00H	Target-related errors	Error at accessed station or relay station.	The corresponding unit is faulty.	Check and correct the error at the designated access destination or the relay station for the access station.

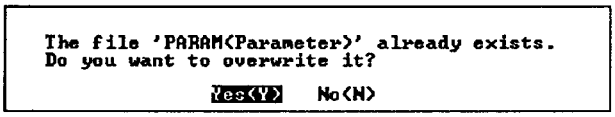
REMARKS

(1) Message (1)

(2) Alert (2)



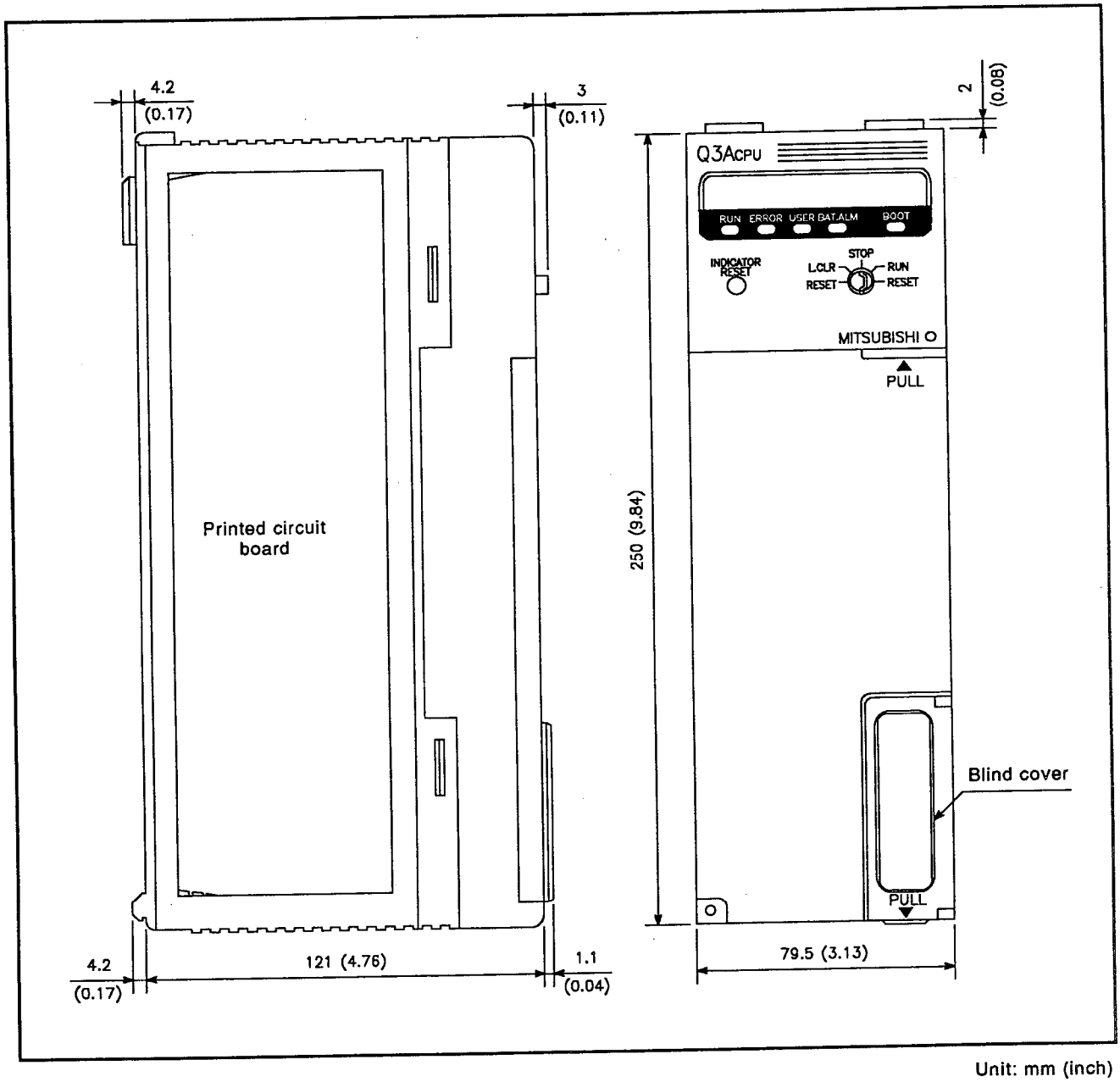
(3) Alert (3)



APPENDIX 6 EXTERNAL DIMENSION DRAWINGS

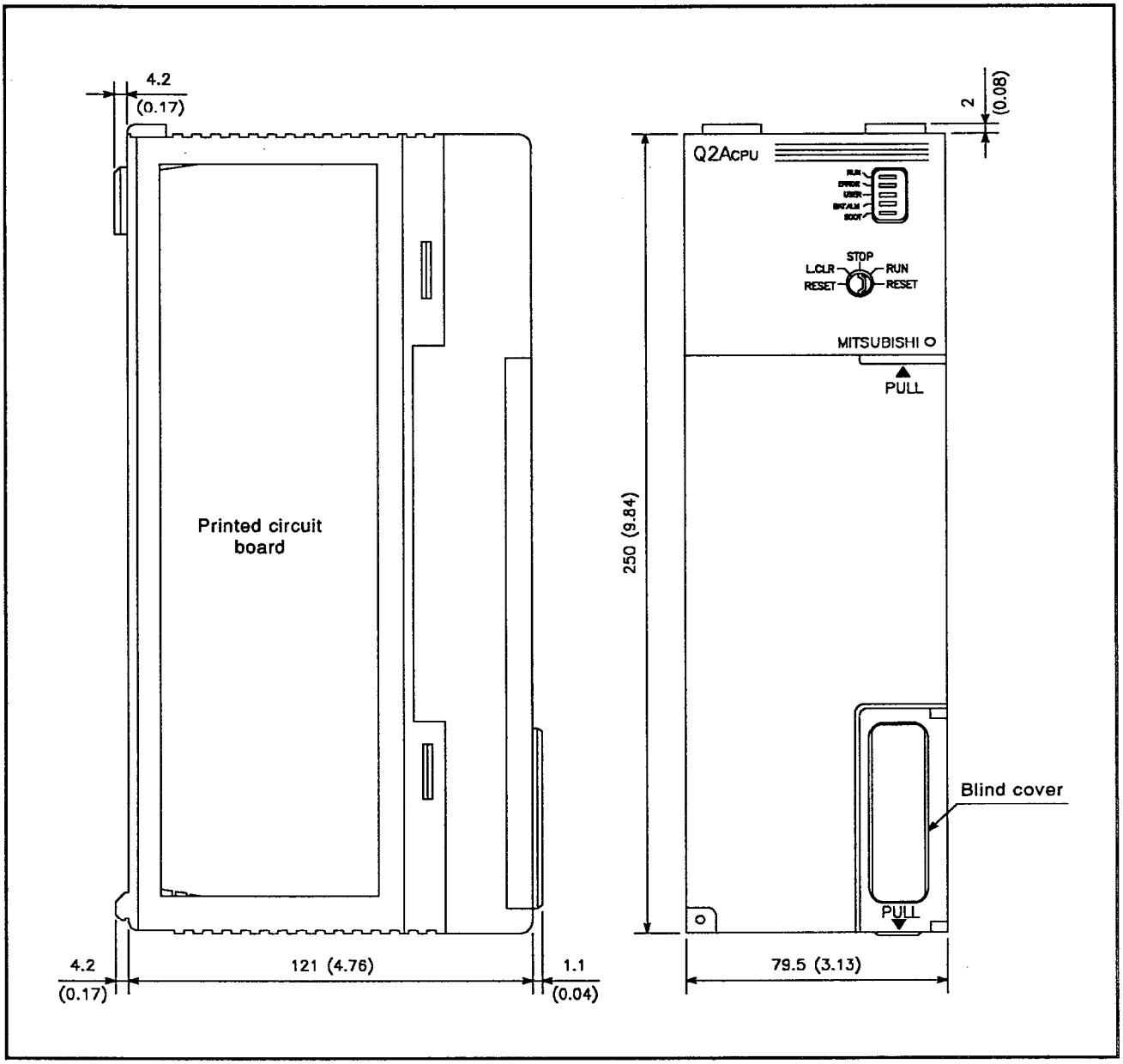
6.1 CPU Modules

(1) Q3ACPU, Q4ACPU modules



Unit: mm (inch)

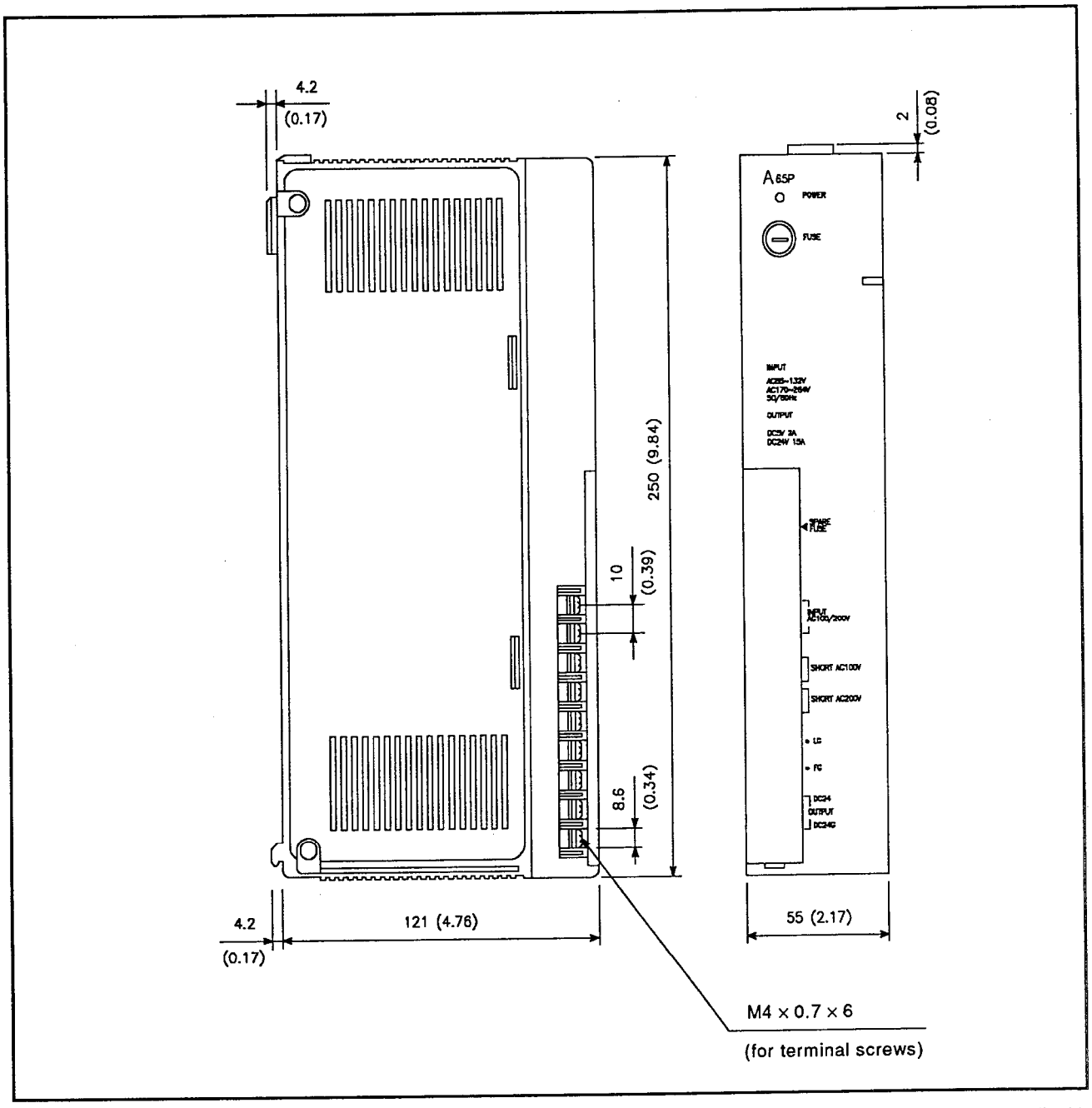
(2) Q2ACPU, Q2ACPU-S1 modules



Unit: mm (inch)

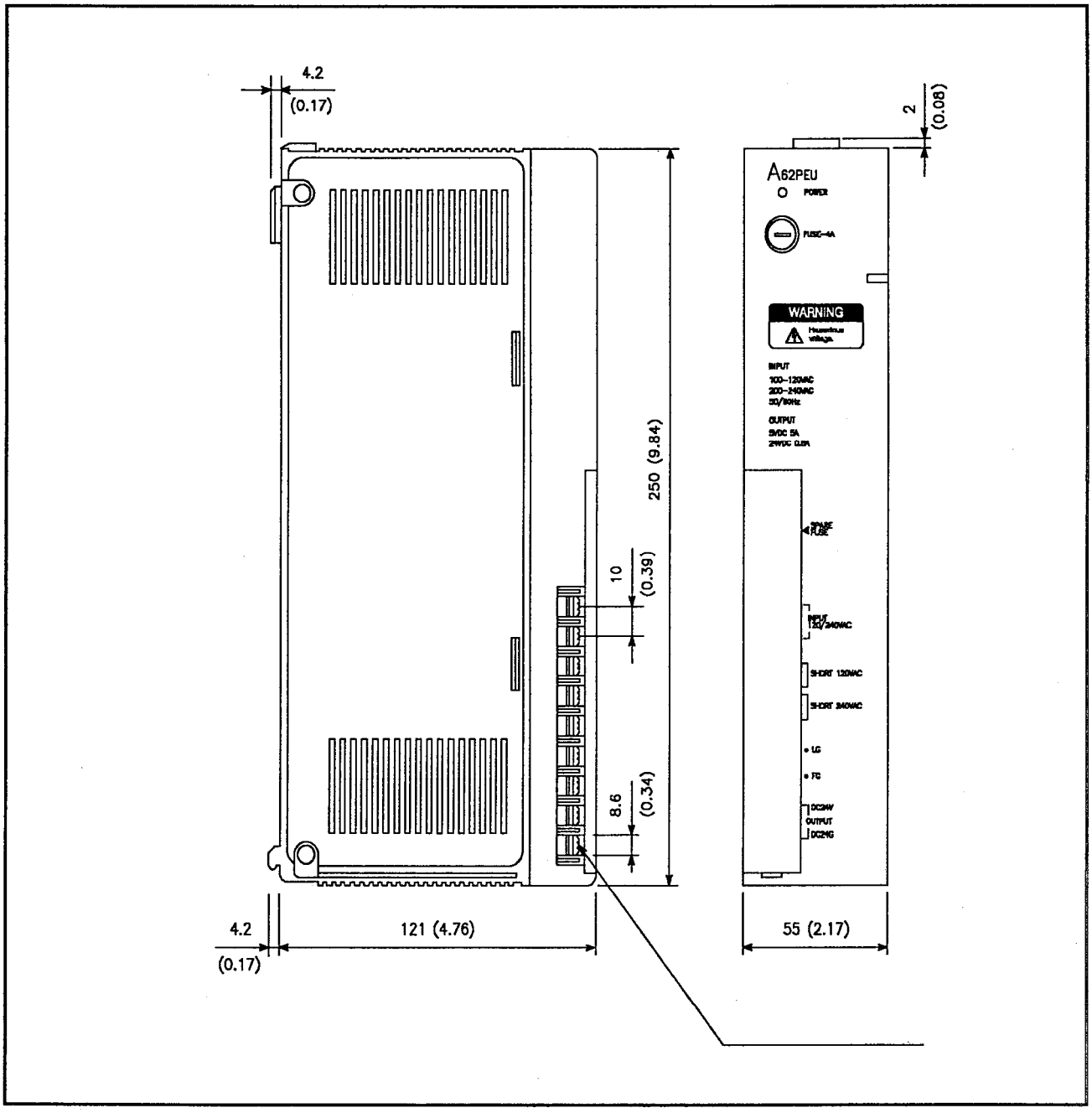
6.2 Power Supply Modules

(1) A61P, A62P, A63P, A65P, A67P power supply modules



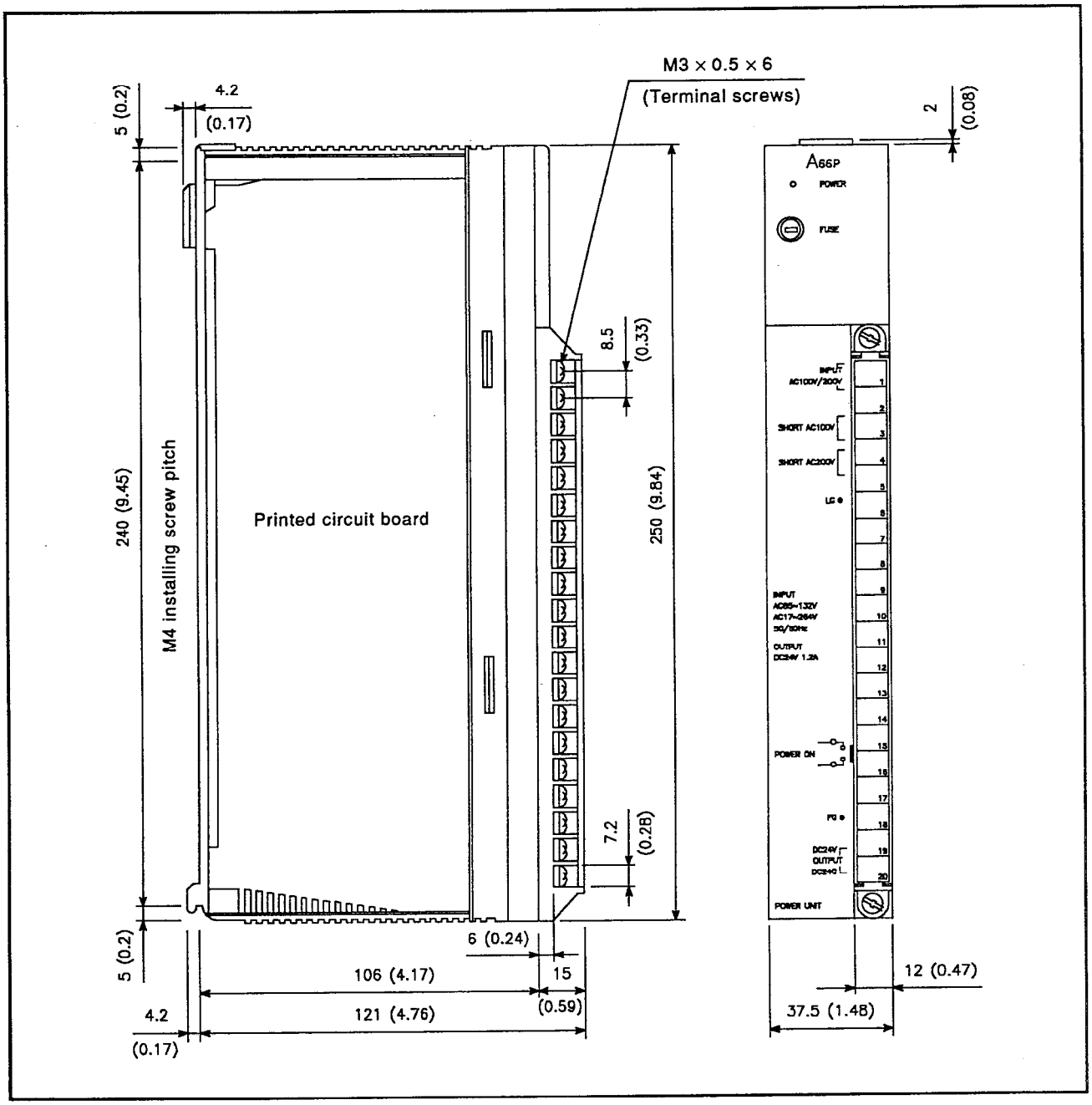
Unit: mm (inch)

(2) A61PEU, A62PEU power supply modules



Unit: mm (inch)

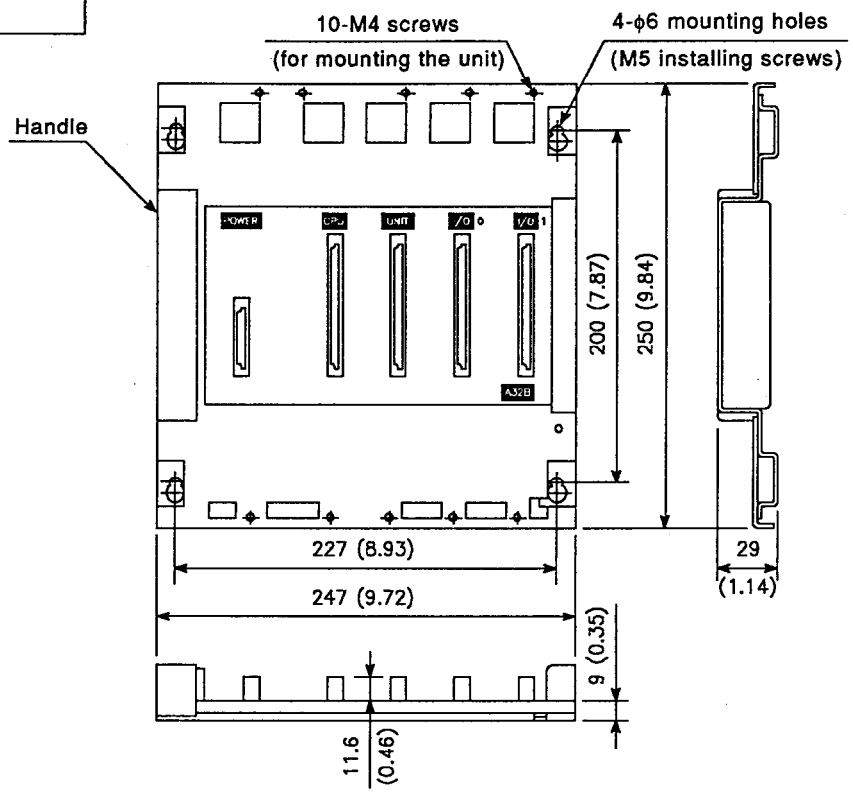
(3) A66P power supply module



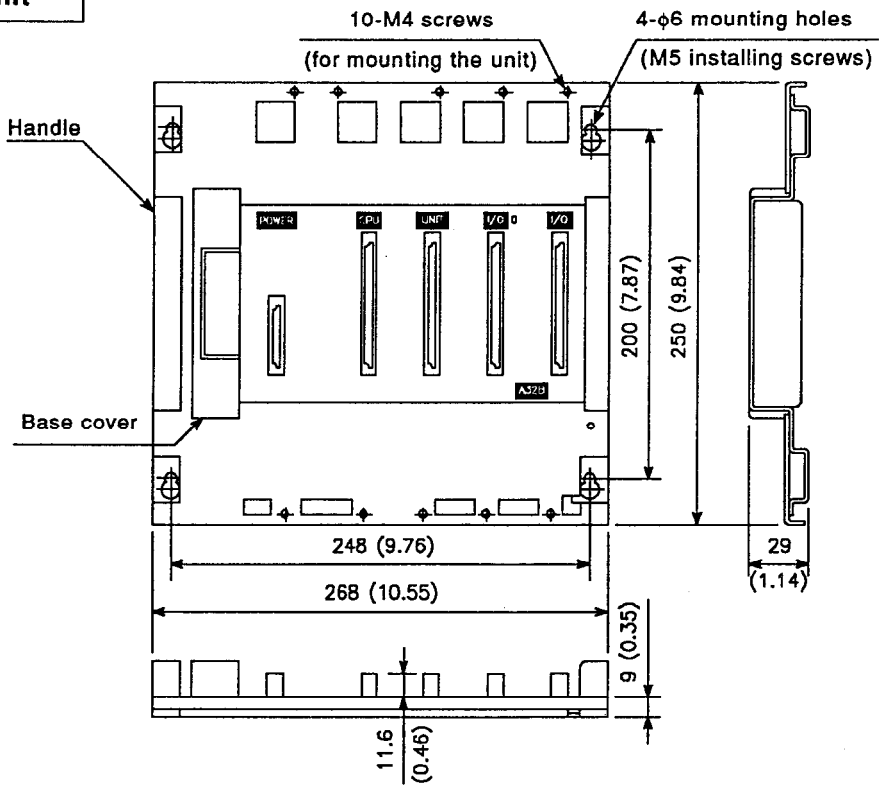
Unit: mm (inch)

6.3 Main Base Units

(1) A32B base unit

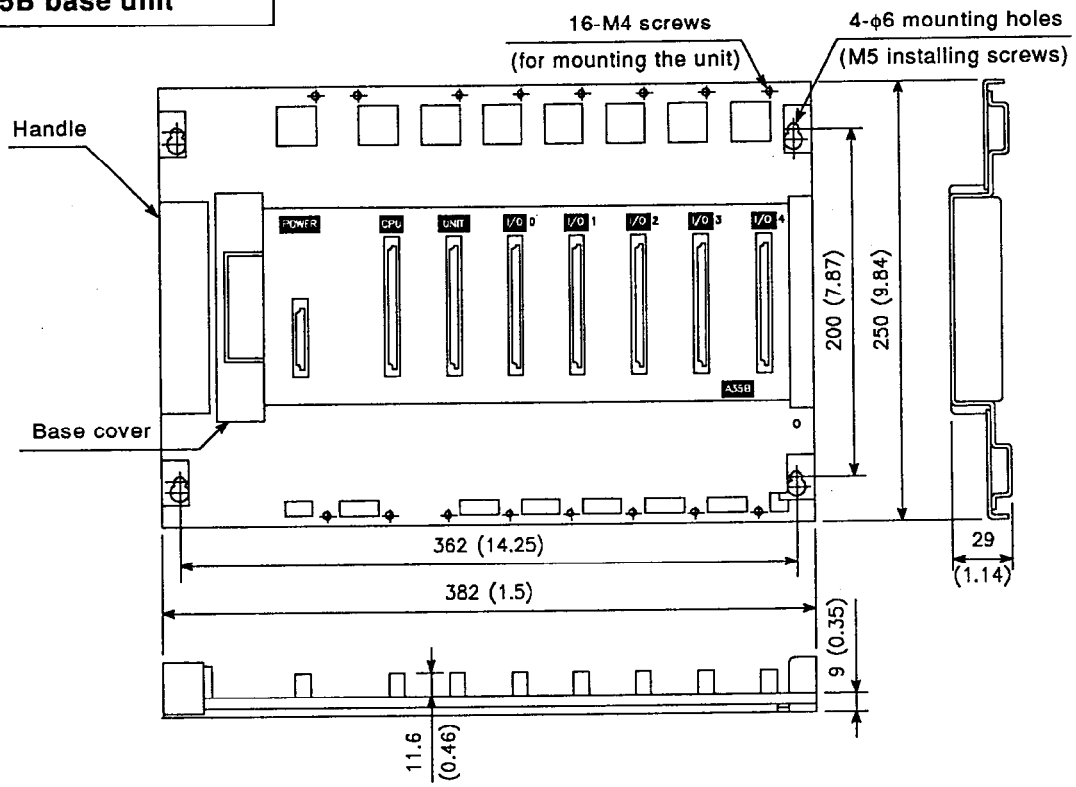


(2) A32B-S1 base unit

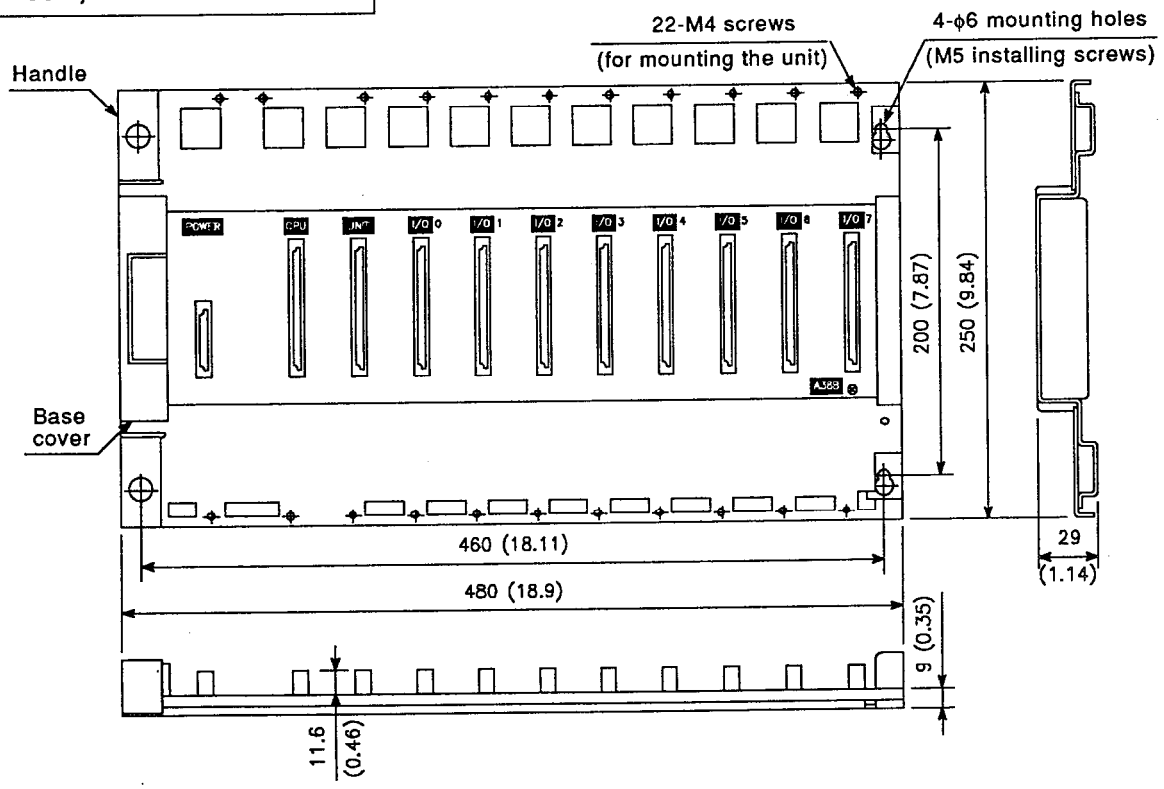


Unit: mm (inch)

(3) A35B base unit



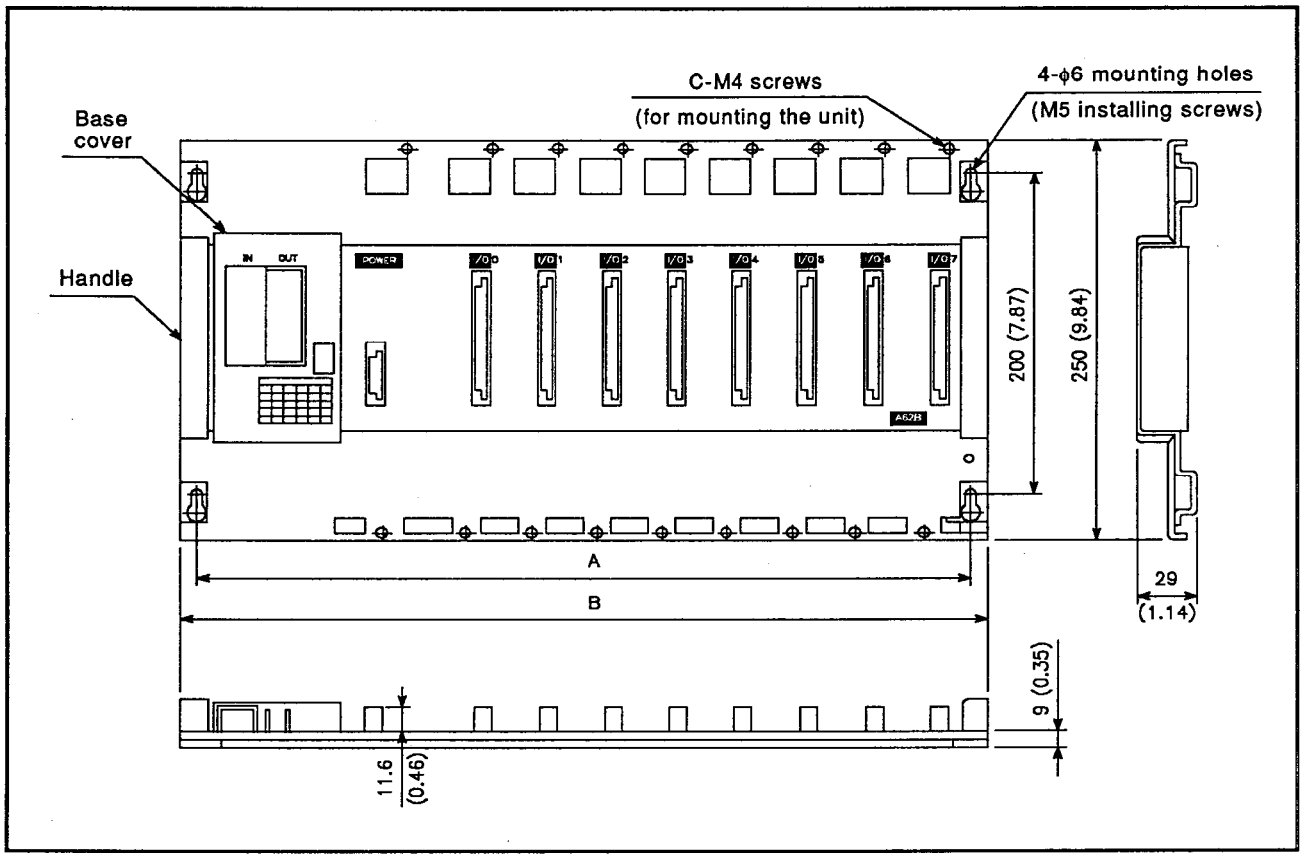
(4) A38B, A38HB base unit



Unit: mm (inch)

6.4 Extension Base Units

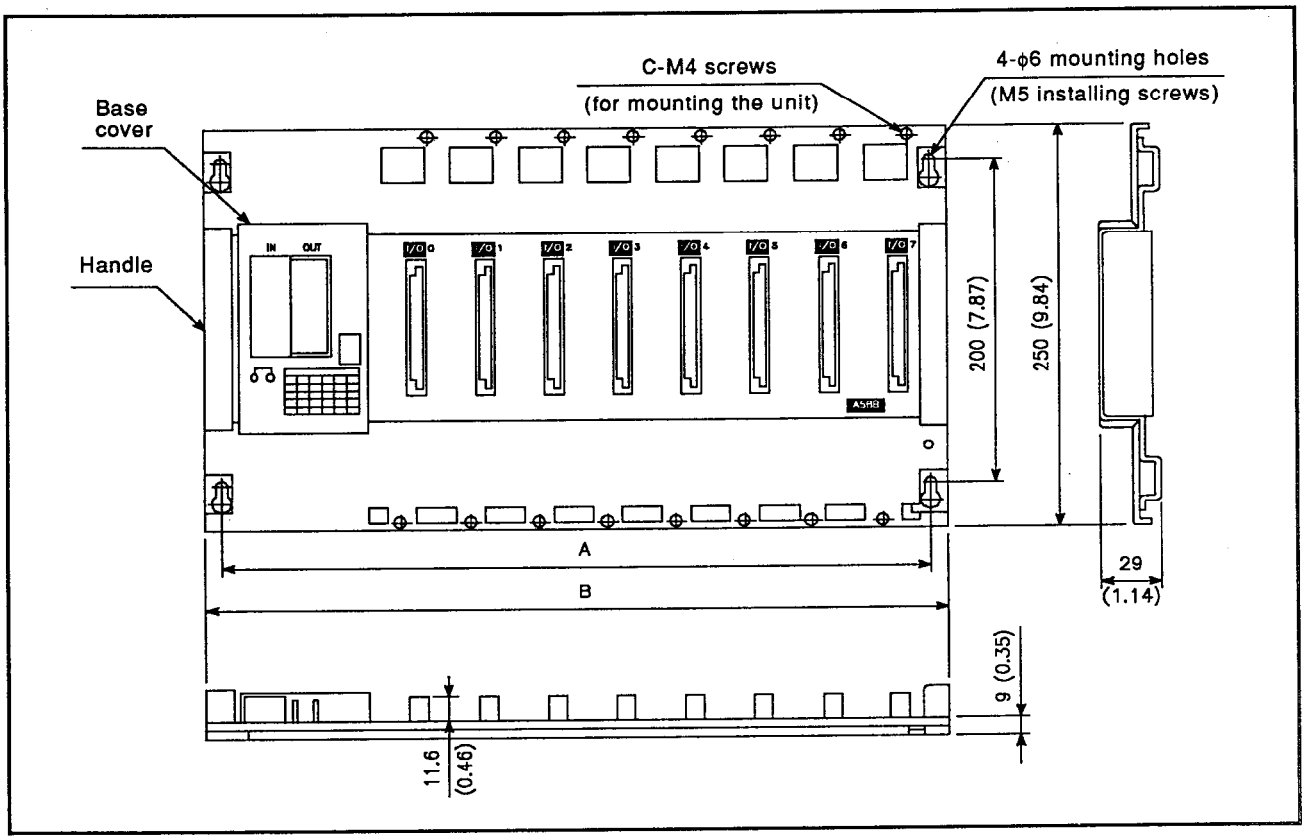
(1) A62B, A65B, A68B base units



Unit: mm (inch)

Model	Variable Dimensions (mm) [inch]			Number of Slots
	A	B	C	
A62B	218 [8.58]	238 [9.37]	6 [0.24]	2 (slots 0, 1)
A65B	332 [13.07]	352 [13.86]	12 [0.47]	5 (slots 0 to 4)
A68B	446 [17.6]	466 [18.35]	18 [0.71]	8 (slots 0 to 7)

(2) A52B, A55B, A58B base units



Unit: mm (inch)

Model	Variable Dimensions (mm) [inch]			Number of Slots
	A	B	C	
A52B	163 [6.42]	183 [7.2]	4 [0.16]	2 (slots 0, 1)
A55B	277 [10.9]	297 [11.69]	10 [0.39]	5 (slots 0 to 4)
A58B	391 [15.4]	411 [16.18]	16 [0.63]	8 (slots 0 to 7)

IMPORTANT

(1) Design the configuration of a system to provide an external protective or safety inter locking circuit for the PCs.

(2) The components on the printed circuit boards will be damaged by static electricity, so avoid handling them directly. If it is necessary to handle them take the following precautions.

(a) Ground your body and the work bench.

(b) Do not touch the conductive areas of the printed circuit board and its electrical parts with non-grounded tools, etc.

Under no circumstances will Mitsubishi Electric be liable or responsible for any consequential damage that may arise as a result of the installation or use of this equipment.

All examples and diagrams shown in this manual are intended only as an aid to understanding the text, not to guarantee operation. Mitsubishi Electric will accept no responsibility for actual use of the product based on these illustrative examples.

Owing to the very great variety in possible applications of this equipment, you must satisfy yourself as to its suitability for your specific application.

HEADQUARTERS

MITSUBISHI ELECTRIC EUROPE B.V. — EUROPE
Gothaer Straße 8
D-40880 Ratingen
Phone: +49 (0) 21 02 / 486-0
Fax: +49 (0) 21 02 / 4 86-11 20
e mail: megfamail@meg.mee.com

MITSUBISHI ELECTRIC EUROPE B.V. — FRANCE
25, Boulevard des Bouvets
F-92741 Nanterre Cedex
Phone: +33 1 55 68 55 68
Fax: +33 1 55 68 56 85
e mail: factory.automation@fra.mee.com

MITSUBISHI ELECTRIC EUROPE B.V. — UK
Travellers Lane
GB-Hatfield Herts. AL10 8 XB
Phone: +44 (0) 1707 / 27 61 00
Fax: +44 (0) 1707 / 27 86 95

MITSUBISHI ELECTRIC EUROPE B.V. — ITALY
Via Paracelso 12
I-20041 Agrate Brianza (MI)
Phone: +39 039 6053 1
Fax: +39 039 6053 312
e mail: factory.automation@it.mee.com

MITSUBISHI ELECTRIC EUROPE B.V. — SPAIN
Carretera de Rubí 76-80
E-08190 Sant Cugat del Vallés
Phone: +34 9 3 / 565 3131
Fax: +34 9 3 / 589 2948
e mail: industrial@sp.mee.com

MITSUBISHI ELECTRIC CORPORATION — JAPAN
Office Tower "Z" 14 F
8-12,1 chome, Harumi Chuo-Ku
Tokyo 104-6212
Phone: +81 3 6221 6060
Fax: +81 3 6221 6075

MITSUBISHI ELECTRIC AUTOMATION — USA
500 Corporate Woods Parkway
Vernon Hills, IL 60061
Phone: +1 847 / 478 21 00
Fax: +1 847 / 478 22 83

EUROPEAN REPRESENTATIVES

GEVA — AUSTRIA
Wiener Straße 89
A-2500 Baden
Phone: +43 (0) 2252 / 85 55 20
Fax: +43 (0) 2252 / 488 60
e mail: office@geva.at

TEHNIKON — BELARUS
Oktjabrskaya 16/5, Ap 704
BY-220030 Minsk
Phone: +375 (0) 17 / 22 75 704
Fax: +375 (0) 17 / 22 76 669
e mail: tehnikon@belsonet.net

Getronics b.v. Control Systems — BELGIUM
Pontbeeklaan 43
B-1731 Asse-Zellik
Phone: +32 (0) 2 / 467 17 51
Fax: +32 (0) 2 / 467 17 45
e mail: infoautomation@getronics.com

TELECON CO. — BULGARIA
4, A. Ljapchev Blvd.
BG-1756 Sofia
Phone: +359 (0) 2 / 97 44 05 8
Fax: +359 (0) 2 / 97 44 06 1
e mail: —

INEA CR d.o.o. — CROATIA
Drvinje 63
HR-10000 Zagreb
Phone: +385 (0) 1 / 36 67 140
Fax: +385 (0) 1 / 36 67 140
e mail: —

AutoCont Control Systems s.r.o. — CZECHIA
Nemocnicni 12
CZ-702 00 Ostrava 2
Phone: +420 59 / 6152 111
Fax: +420 59 / 6152 562
e mail: consys@autocont.cz

louis poulsen industri & automation — DENMARK
Geminivej 32
DK-2670 Greve
Phone: +45 (0) 43 / 95 95 95
Fax: +45 (0) 43 / 95 95 91
e mail: lpia@lpmail.com

UTU Elektrotehnika AS — ESTONIA
Pärnu mnt.160i
EE-11317 Tallinn
Phone: +372 (0) 6 / 51 72 80
Fax: +372 (0) 6 / 51 72 88
e mail: utu@utu.ee

Beijer Electronics OY — FINLAND
Ansatie 6a
FIN-01740 Vantaa
Phone: +358 (0) 9 / 886 77 500
Fax: +358 (0) 9 / 886 77 555
e mail: info@beijer.fi

PROVENDOR OY — FINLAND
Teljänkatu 8 A 3
FIN-28130 Pori
Phone: +358 (0) 2 / 522 3300
Fax: +358 (0) 2 / 522 3322
e mail: —

UTECO A.B.E.E. — GREECE
5, Mavrogenous Str.
GR-18542 Piraeus
Phone: +302 (0) 10 / 42 10 050
Fax: +302 (0) 10 / 42 12 033
e mail: uteco@uteco.gr

Meltrade Automatika Kft. — HUNGARY
55, Harmat St.
H-1105 Budapest
Phone: +36 (0)1 / 2605 602
Fax: +36 (0)1 / 2605 602
e mail: office@meltrade.hu

EUROPEAN REPRESENTATIVES

MITSUBISHI ELECTRIC EUROPE B.V. – Irish Branch — IRELAND
Westgate Business Park
IRL-Dublin 24
Phone: +353 (0) 1 / 419 88 00
Fax: +353 (0) 1 / 419 88 90
e mail: sales.info@meir.mee.com

SIA POWEL — LATVIA
Lienes iela 28
LV-1009 Riga
Phone: +371 784 / 22 80
Fax: +371 784 / 22 81
e mail: utu@utu.lv

UAB UTU POWEL — LITHUANIA
Savanoriu pr. 187
LT-2053 Vilnius
Phone: +370 (0) 52323-101
Fax: +370 (0) 52322-980
e mail: powel@utu.lt

INTEHSIS SRL — MOLDOVA, REPUBLIC OF
Cuza-Voda 36/1-81
MD-2061 Chisinau
Phone: +373 (0)2 / 562 263
Fax: +373 (0)2 / 562 263
e mail: intehsis@mdl.net

Getronics b.v. Control Systems — NETHERLANDS
Donauweg 2 B
NL-1043 AJ Amsterdam
Phone: +31 (0) 20 / 587 67 00
Fax: +31 (0) 20 / 587 68 39
e mail: info.gia@getronics.com

Beijer Electronics AS — NORWAY
Teglverksveien 1
N-3002 Drammen
Phone: +47 (0) 32 / 24 30 00
Fax: +47 (0) 32 / 84 85 77
e mail: info@beijer.no

MPL Technology Sp. z o.o. — POLAND
ul. Sliczna 36
PL-31-444 Kraków
Phone: +48 (0) 12 / 632 28 85
Fax: +48 (0) 12 / 632 47 82
e mail: krakow@mpl.pl

Sirius Trading & Services srl — ROMANIA
Bd. Lacul Tei nr. 1 B
RO-72301 Bucuresti 2
Phone: +40 (0) 21 / 201 7147
Fax: +40 (0) 21 / 201 7148
e mail: sirius_t_s@fx.ro

ACP Autocomp a.s. — SLOVAKIA
Chalupkova 7
SK-81109 Bratislava
Phone: +421 (02) / 5292-22 54, 55
Fax: +421 (02) / 5292-22 48
e mail: info@acp-autocomp.sk

INEA d.o.o. — SLOVENIA
Stegne 11
SI-1000 Ljubljana
Phone: +386 (0) 1-513 8100
Fax: +386 (0) 1-513 8170
e mail: inea@inea.si

Beijer Electronics AB — SWEDEN
Box 426
S-20124 Malmö
Phone: +46 (0) 40 / 35 86 00
Fax: +46 (0) 40 / 35 86 02
e mail: info@beijer.se

ECONOTEC AG — SWITZERLAND
Postfach 282
CH-8309 Nürensdorf
Phone: +41 (0) 1 / 838 48 11
Fax: +41 (0) 1 / 838 48 12
e mail: info@econotec.ch

EUROPEAN REPRESENTATIVES

GTS — TURKEY
Darülaceze Cad. No. 43 KAT: 2
TR-80270 Okmeydani-Istanbul
Phone: +90 (0) 212 / 320 1640
Fax: +90 (0) 212 / 320 1649
e mail: gts@turk.net

CSC Automation Ltd. — UKRAINE
15, M. Raskova St., Fl. 10, Office 1010
UA-02002 Kiev
Phone: +380 (0) 44 / 238-83-16
Fax: +380 (0) 44 / 238-83-17
e mail: csc-a@csc-a.kiev.ua

AFRICAN REPRESENTATIVE

CBI Ltd — SOUTH AFRICA
Private Bag 2016
ZA-1600 Isando
Phone: +27 (0) 11/ 928 2000
Fax: +27 (0) 11/ 392 2354
e mail: cbi@cbi.co.za

MIDDLE EAST REPRESENTATIVE

TEXEL Electronics LTD. — ISRAEL
Box 6272
IL-42160 Netanya
Phone: +972 (0) 9 / 863 08 91
Fax: +972 (0) 9 / 885 24 30
e mail: texel_me@netvision.net.il

EURASIAN REPRESENTATIVE

AVTOMATIKA SEVER — RUSSIA
Krapivnij Per. 5, Of. 402
RU-194044 St Petersburg
Phone: +7 812 / 1183 238
Fax: +7 812 / 3039 648
e mail: pav@avtsev.spb.ru

CONSYS — RUSSIA
Promyshlennaya St. 42
RU-198099 St Petersburg
Phone: +7 812 / 325 36 53
Fax: +7 812 / 325 36 53
e mail: consys@consys.spb.ru

ELEKTROSTYLE — RUSSIA
Ul Garschina 11
RU-140070 Moscowskaja Oblast
Phone: +7 095/ 261 3808
Fax: +7 095/ 261 3808
e mail: —

ICOS — RUSSIA
Industrial Computer Systems Zao
Ryazanskij Prospekt 8a, Office 100
RU-109428 Moscow
Phone: +7 095 / 232 - 0207
Fax: +7 095 / 232 - 0327
e mail: mail@icos.ru

NPP Uralelektra — RUSSIA
Sverdlova 11a
RU-620027 Ekaterinburg
Phone: +7 34 32 / 53 27 45
Fax: +7 34 32 / 53 27 45
e mail: elektra@etel.ru

STC Drive Technique — RUSSIA
Poslannikov Per. 9, str.1
RU-107005 Moscow
Phone: +7 095 / 786 21 00
Fax: +7 095 / 786 21 01
e mail: info@privod.ru